Second Edition

The Human-Computer Interaction Handbook

Fundamentals, Evolving Technologies and Emerging Applications



Human Factors and Ergonomics

Edited by

Andrew Sears

Julie A. Jacko

The Human–Computer Interaction Handbook

Fundamentals, Evolving Technologies, and Emerging Applications

Second Edition

HUMAN FACTORS AND ERGONOMICS

Gavriel Salvendy, Series Editor

Aykin, N. (Ed.): *Usability and Internationalization of Information Technology.*

Bagnara, S. (Ed.): *Theories and Practice in Interaction Design*

Buck, J. R., and Lehto, M. R. (Au.): *Introduction to Human Factors and Ergonomics for Engineers*

Carayon, P. (Ed.): *Handbook of Human Factors and Ergonomics in Health Care and Patient Safety*

Hendrick, H., and Kleiner, B. (Eds.): *Macroergonomics: Theory, Methods and Applications.*

Hollnagel, E. (Ed.): *Handbook of Cognitive Task Design.*

Jacko, J. A., and Sears, A. (Eds.): *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.*

Karwowski, W. (Ed.): *Handbook of Standards and Guidelines in Ergonomics and Human Factors.*

Keates, S. (Au.): *Designing for Accessibility: A Business Guide to Countering Design Exclusion*

Meister, D. (Au.): *Conceptual Foundations of Human Factors Measurement.*

Meister, D., and Enderwick, T. (Eds.): *Human Factors in System Design, Development, and Testing.*

Proctor, R., and Vu, K. (Eds.): *Handbook of Human Factors in Web Design.*

Sears, A., and Jacko, J. A. (Eds.): *Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Second Edition.*

Stanney, K. (Ed.): *Handbook of Virtual Environments: Design, Implementation, and Applications.*

Stephanidis, C. (Ed.): *User Interfaces for All: Concepts, Methods, and Tools.*

Wogalter, M. (Eds.): *Handbook of Warnings.*

Ye, N. (Ed.): *The Handbook of Data Mining.*

Also in this Series

HCI International 1999 Proceedings **2 Volume Set**

HCI International 2001 Proceedings **3 Volume Set**

HCI International 2003 Proceedings **4 Volume Set**

HCI International 2005 Proceedings **11 Volume CD Rom Set ISBN# 0-8058-5807-5**

The Human–Computer Interaction Handbook

Fundamentals, Evolving Technologies, and Emerging Applications

Second Edition

Edited by

Andrew Sears
UMBC

and

Julie A. Jacko
Georgia Institute of Technology

Lawrence Erlbaum Associates
Taylor & Francis Group
New York London

Lawrence Erlbaum Associates
Taylor & Francis Group
270 Madison Avenue
New York, NY 10016

Lawrence Erlbaum Associates
Taylor & Francis Group
2 Park Square
Milton Park, Abingdon
Oxon OX14 4RN

© 2008 by Taylor & Francis Group, LLC
Lawrence Erlbaum Associates is an imprint of Taylor & Francis Group, an Informa business

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-0-8058-5870-9 (Hardcover)

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the LEA Web site at
http://www.erlbaum.com

For Beth, Nicole, Kristen, François, and Nicolas.

CONTENTS

Contributors xi

Advisory Board xv

Series Foreword (*Gavriel Salvendy*) xvii

Foreword (*Ben Shneiderman*) xix

Preface xxi

About the Editors xxiii

Introduction

A Moving Target: The Evolution of HCI 1
Jonathan Grudin

PART I—Humans in HCI 25

1 Perceptual-Motor Interaction: Some Implications for HCI 27
Timothy N. Welsh, Romeo Chua, Daniel J. Weeks, and David Goodman

2 Human Information Processing: An Overview for Human-Computer Interaction 43
Robert W. Proctor and Kim-Phuong L. Vu

3 Mental Models in Human-Computer Interaction 63
Stephen J. Payne

4 Emotion in Human-Computer Interaction 77
Scott Brave and Cliff Nass

5 Cognitive Architecture 93
Michael D. Byrne

6 Task Loading and Stress in Human-Computer Interaction: Theoretical Frameworks and Mitigation Strategies 115
J. L. Szalma and Peter Hancock

7 Motivating, Influencing, and Persuading Users: An Introduction to Captology 133
B. J. Fogg, Gregory Cueller, and David Danielson

8 Human-Error Identification in Human-Computer Interaction 147
Neville Stanton

Part II—Computers in HCI 159

9 Input Technologies and Techniques 161
Ken Hinckley

10 Sensor- and Recognition-Based Input for Interaction 177
Andrew D. Wilson

11 Visual Displays 201
Christopher Schlick, Martina Ziefle, Milda Park, and Holger Luczak

12 Haptic Interfaces 229
Hiroo Iwata

13 Nonspeech Auditory Output 247
Stephen Brewster

14 Network-Based Interaction 265
Alan Dix

15 Wearable Computers 295
Dan Siewiorek, Asim S. Mailagic, and Thad Starner

16 Design of Computer Workstations 313
Michael J. Smith, Pascale Carayon, and William J. Cohen

Part III—Designing Human-Computer Interactions 327

17 Visual Design: Principles for Usable Interfaces: Everything Is Designed: Why We Should Think Before Doing 329
Suzanne Watzman and Margaret Re

18 Global/Intercultural User Interface Design 355
Aaron Marcus

19 Conversational Speech Interfaces and Technologies 381
Jennifer Lai, Clare-Marie Karat, and Nicole Yankelovich

20 Multimedia User Interface Design 393
Alistair Sutcliffe

21 Multimodal Interfaces 413
Sharon Oviatt

22 Adaptive Interfaces and Agents 433
Anthony Jameson

23 Mobile Interaction Design in the Age of Experience Ecosystems 459
Marco Susani

24 Tangible User Interfaces 469
Hiroshi Ishii

25 Achieving Psychological Simplicity: Measures and Methods to Reduce Cognitive Complexity 489
John C. Thomas and John T. Richards

26 Information Visualization 509
Stuart Card

27 Groupware and Computer-Supported Cooperative Work 545
Gary M. Olson and Judith S. Olson

28 HCI and the Web 559
Helen Ashman, Tim Brailsford, Gary Burnett, Jim Goulding, Adam Moore, Craig Stewart, and Mark Truran

29 Human-Centered Design of Decision-Support Systems 573
Philip J. Smith, Norman D. Geddes, and Roger Beatty

30 Online Communities 603
Panayiotis Zaphiris, Chee Siang Ang, and Andrew Laghos

31 Virtual Environments 621
Kay M. Stanney and Joseph V. Cohn

32 Human-Computer Interaction Viewed from the Intersection of Privacy, Security, and Trust 639
John Karat, Clare-Marie Karat, and Carolyn Brodie

Part IV—Application/Domain Specific Design 559

33 Human-Computer Interaction in Health Care
François Sainfort, Julie Jacko, Paula J. Edwards, and Bridget C. Booske 661

34 Why We Play: Affect and the Fun of Games: Designing Emotions for Games, Entertainment Interfaces and Interactive Products 679
Nicole Lazzaro

35 Motor Vehicle Driver Interfaces 701
Paul Green

36 Human-Computer Interaction in Aerospace 721
Steven J. Landry

37 User-Centered Design in Games 741
Randy J. Pagulayan, Kevin Keeker, Thomas Fuller, Dennis Wixon, and Ramon L. Romero

Part V—Designing for Diversity 761

38 The Digital Divide: The Role of Gender in Human-Computer Interaction 763
Joel Cooper and Matthew B. Kugler

39 Information Technology and Older Adults 777
Sara J. Czaja and Chin Chin Lee

40 HCI for Kids 793
Amy Bruckman, Alisa Bandlow, and Andrea Forte

41 Information Technology for Cognitive Support 811
A. F. Newell, A. Carmichael, P. Gregor, N. Alm, and A. Waller

42 Physical Disabilities and Computing Technologies: An Analysis of Impairments 829
Andrew Sears, Mark Young, and Jinjuan Feng

43 Perceptual Impairments: New Advancements Promoting Technological Access 853
Julie A. Jacko, V. Kathlene Leonard, and Ingrid Scott

44 Universal Accessibility and Functionally Illiterate Populations: Implications for HCI, Design, and Testing 871
William M. Gribbons

45 Computing Technologies for Deaf and Hard of Hearing Users 885
Vicki L. Hanson

Part VI—The Development Process 895

Section A: Requirements Specification

46 User Experience and HCI 897
Mike Kuniavsky

47 Requirements Specifications within the Usability Engineering Lifecycle 917
Deborah J. Mayhew

48 Task Analysis 927
Catherine Courage, Janice (Ginny) Redish, and Dennis Wixon

49 Contextual Design 949
Karen Holtzblatt

50 The Ethnographic Approach to Design 965
Jeanette Blomberg, Mark Burrell

Section B: Design and Development

51 Putting Personas to Work: Using Data-Driven Personas to Focus Product Planning, Design and Development 991
Tamara Adlin and John Pruitt

52 Prototyping Tools and Techniques 1017
Michel Beaudouin-Lafon and Wendy E. Mackay

53 Scenario-based Design 1041
Mary Beth Rosson and John M. Carroll

54 Participatory Design: The Third Space in HCI 1061
Michael J. Muller

55 Unified User Interface Development: New Challenges and Oppurtunities 1083
Anthony Savidis and Constantine Stephanidis

56 HCI and Software Engineering: Designing for User Interface Plasticity 1107
Jöelle Coutaz and Gäelle Calvary

Section C: Testing and Evaluation

57 Usability Testing: Current Practice and Future Directions 1129
Joseph S. Dumas, and Jean E. Fox

58 Survey Design and Implementation in HCI 1151
A. Ant Ozok

59 Inspection-based Evaluations 1171
Gilbert Cockton, Alan Woolrych, and Darryn Lavery

60 Model-Based Evaluation 1191
David Kieras

Part VII—Managing HCI and Emerging Issues 1209

61 Technology Transfer 1211
Kevin M. Schofield

62 Augmenting Cognition in HCI: 21st Century Adaptive System Science and Technology 1221
Dylan Schmorrow, Amy Kruse, Leah Reeves, and Amy Bolton

63 Human Values, Ethics, and Design 1241
Batya Friedman and Peter H. Kahn, Jr.

64 Cost Justification 1267
J. Patrick Williams, Randolph G. Bias, and Deborah J. Mayhew

Perspectives on HCI 1279

65 Future Trends in Human-Computer Interaction 1281
Andrew Sears and Julie Jacko

Author Index 1291

Subject Index 1347

CONTRIBUTORS

Tamara Adlin
adlin inc., USA

Norman Alm
School of Computing, University of Dundee, Scotland

Chee Siang Ang
Centre for HCI Design, City University London, UK

Helen Ashman
School of Computer Science and Information Technology, University of Nottingham, UK

Alisa Bandlow
College of Computing, Georgia Institute of Technology, USA

Roger Beatty
American Airlines, USA

Michel Beaudouin-Lafon
Université Paris—Sud, France

Randolph G. Bias
School of Information, The University of Texas at Austin, USA

Jeanette Blomberg
IBM Almaden Research Center, USA

Amy Bolton
Strategic Analysis, USA

Bridget C. Booske
Population Health Institute, School of Medicine and Public Health, University of Wisconsin, USA

Tim Brailsford
School of Computer Science and Information Technology, University of Nottingham, UK

Scott Brave
Baynote Inc., USA

Stephen Brewster
Department of Computing Science, University of Glasgow, UK

Carolyn Brodie
IBM T.J. Watson Research Center, USA

Amy Bruckman
College of Computing, Georgia Institute of Technology, USA

Gary Burnett
School of Computer Science and Information Technology, University of Nottingham, UK

Mark Burrell
Microsoft, USA

Michael D. Byrne
Department of Psychology, Rice University, USA

Gaëlle Calvary
Université Joseph Fourier, France

Pascale Carayon
Department of Industrial Engineering, University of Wisconsin-Madison, USA

Stuart Card
User Interface Research Group, Palo Alto Research Center (PARC), USA

Alex Carmichael
Department of Psychology, Manchester University, UK

John M. Carroll
College of Information Sciences and Technology, The Pennsylvania State University, USA

Romeo Chua
School of Human Kinetics, University of British Columbia, Canada

Gilbert Cockton
School of Computing and Technology, University of Sunderland, UK

William Cohen
Exponent Failure Analysis Associates, USA

Joseph V. Cohn
Naval Research Laboratory, USA

Joel Cooper
Department of Psychology, Princeton University, USA

Catherine Courage
salesforce.com, USA

Joëlle Coutaz
Université Joseph Fourier, France

Gregory Cuellar
Communication Department, Stanford University, USA

Sara J. Czaja
Department of Psychiatry and Behavioral Sciences, Center on Aging, University of Miami Miller School of Medicine, USA

David Danielson
Communication Department, Stanford University, USA

Alan Dix
Computing Department, Lancaster University, UK

Joseph S. Dumas
Bentley College, USA

Paula J. Edwards
Health Systems Institute, Georgia Institute of Technology, USA

Jinjuan Feng
Department of Computer and Information Sciences, Towson University, USA

B. J. Fogg
Persuasive Technology Lab, Center for the Study of Language and Information, Stanford University, USA

Andrea Forte
College of Computing, Georgia Institute of Technology, USA

Jean E. Fox
Bureau of Labor Statistics, USA

Batya Friedman
The Information School, University of Washington, USA

Thomas Fuller
Microsoft Game Studios, Microsoft Corporation, USA

Norman D. Geddes
Applied Systems Intelligence, Inc., USA

David Goodman
School of Kinesiology, Simon Fraser University, Canada

Jim Goulding
School of Computer Science and Information Technology, University of Nottingham, UK

Paul Green
Human Factors Division, University of Michigan, Transportation Research Institute, USA

Peter Gregor
School of Computing, University of Dundee, Scotland

William Gribbons
Bentley College, USA

Jonathan Grudin
Microsoft Corporation, USA

P. A. Hancock
Department of Psychology, and The Institute for Simulation and Training, University of Central Florida, USA

Vicki L. Hanson
IBM T.J. Watson Research Center, USA

Ken Hinckley
Microsoft Research, USA

Karen Holtzblatt
InContext Enterprises, Inc., USA

Hiroshi Ishii
MIT Media Lab, USA

Hiroo Iwata
Graduate School of Systems and Information Engineering, University of Tsukuba, Japan

Julie A. Jacko
Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology and Emory University School of Medicine, USA

Anthony Jameson
German Research Center for Artificial Intelligence, and International University in Germany, Germany

Peter H. Kahn, Jr.
Department of Psychology, University of Washington, USA

Clare-Marie Karat
IBM T.J. Watson Research Center, USA

John Karat
IBM T.J. Watson Research Center, USA

Kevin Keeker
Microsoft Game Studios, Microsoft Corporation, USA

David Kieras
Electrical Engineering and Computer Science Department, University of Michigan, USA

Amy Kruse
Defense Advanced Research Projects Agency, USA

Matthew Kugler
Department of Psychology, Princeton University, USA

Mike Kuniavsky
ThingM, USA

Andrew Laghos
Centre for HCI Design, City University London, UK

Jennifer Lai
IBM T.J. Watson Research Center, USA

Steven Landry
Department of Industrial Engineering, Purdue University, USA

Darryn Lavery
Microsoft Corporation, USA

Nicole Lazzaro
XEODesign, Inc., USA

Chin Chin Lee
Center on Aging, University of Miami, USA

V. Kathlene Leonard
Alucid Solution, Inc., USA

Holger Luczak
Institute of Industrial Engineering and Ergonomics, RWTH Aachen University, Germany

Wendy E. Mackay
INRIA, France

Asim S. Mailagic
College of Engineering, Carnegie-Mellon University, USA

Aaron Marcus
Aaron Marcus and Associates, Inc., USA

Deborah J. Mayhew
Deborah J. Mayhew and Associates, USA

Adam Moore
School of Computer Science and Information Technology, University of Nottingham, UK

Michael J. Muller
IBM Research, USA

Clifford Nass
Department of Communication, Stanford University, USA

Alan F. Newell
School of Computing, University of Dundee, Scotland

Gary M. Olson
School of Information, The University of Michigan, USA

Judith S. Olson
School of Information, The University of Michigan, USA

Katsuhiko Ogawa
Department of Research and Development, Nippon Telegraph and Telephone Corporation, Japan

Sharon Oviatt
Department of Computer Science and Engineering, Oregon Health and Science University, USA

A. Ant Ozok
Department of Information Systems, UMBC, USA

Randy J. Pagulayan
Microsoft Game Studios, Microsoft Corporation, USA

Milda Park
Institute of Industrial Engineering and Ergonomics, RWTH Aachen University, Germany

Stephen J. Payne
University of Manchester, UK

Robert W. Proctor
Department of Psychological Sciences, Purdue University, USA

John Pruitt
Microsoft Corporation, USA

P. L. Patrick Rau
Department of Industrial Engineering, Tsinghua University, China

Peggy Re
Visual Arts Department, UMBC, USA

Leah Reeves
Potomac Institute for Policy Studies, USA

Janice (Ginny) Redish
Redish & Associates, Inc., USA

John T. Richards
IBM T. J. Watson Research Center, USA

Ramon L. Romero
Microsoft Game Studios, Microsoft Corporation, USA

Mary Beth Rosson
College of Information Sciences and Technology, The Pennsylvania State University, USA

François Sainfort
Health Systems Institute, Georgia Institute of Technology, USA

Gavriel Salvendy
Department of Industrial Engineering, Purdue University, USA, and Department of Industrial Engineering, Tsinghua University, China

Anthony Savidis
Institute of Computer Science, Foundation for Research and Technology—Hellas (ICS-FORTH), Greece

Christopher Schlick
Institute of Industrial Engineering and Ergonomics, RWTH Aachen University, Germany

Dylan Schmorrow
Office of Naval Research, USA

Kevin Schofield
Microsoft Corporation, USA

Ingrid U. Scott
Department of Ophthalmology, Penn State College of Medicine, USA

Andrew Sears
Interactive Systems Research Center, Information Systems Department, UMBC, USA

Daniel P. Siewiorek
Human-Computer Interaction Institute, Carnegie-Mellon University, USA

Michael J. Smith
Department of Industrial Engineering, University of Wisconsin-Madison, USA

Philip J. Smith
Institute for Ergonomics, Ohio State University, USA

Kay M. Stanney
Industrial Engineering and Management Systems, University of Central Florida, USA

Neville A. Stanton
School of Engineering and Design, Brunel University, UK

Thad Starner
College of Computing, Georgia Institute of Technology, USA

Constantine Stephanidis
Institute of Computer Science, Foundation for Research and Technology—Hellas (ICS-FORTH), and Department of Computer Science, University of Crete, Greece

Craig Stewart
School of Computer Science and Information Technology, University of Nottingham, and Department of Electronic Engineering, Queen Mary, University of London, UK

Marco Susani
Design for Seamless Mobility, Motorola, USA

Alistair Sutcliffe
School of Informatics, University of Manchester, UK

J. L. Szalma
Department of Psychology, University of Central Florida, USA

John C. Thomas
IBM T. J. Watson Research Center, USA

Mark Truran
University of Teesside, UK

Kim-Phuong L. Vu
Department of Psychology, California State University Long Beach, USA

Annalu Waller
School of Computing, University of Dundee, Scotland

Suzanne Watzman
Watzman Design, USA

Daniel J. Weeks
Department of Psychology, Simon Fraser University, Canada

Timothy N. Welsh
Faculty of Kinesiology, University of Calgary, Canada

J. Patrick Williams
School of Information, The University of Texas at Austin, USA

Andrew Wilson
Microsoft Research, USA

Dennis Wixon
Microsoft Game Studios, Microsoft Corporation, USA

Alan Woolrych
School of Computing and Technology, University of Sunderland, UK

Nicole Yankelovich
Sun Microsystems, USA

Mark A. Young
The Maryland Rehabilitation Center and Workforce Technology Center, State of Maryland Department of Education, USA

Panayiotis Zaphiris
Centre for HCI Design, City University London, UK

Martina Ziefle
Institute for Psychology, RWTH Aachen University, Germany

ADVISORY BOARD

Noëlle Carbonell
University Henri Poincaré–Nancy 1, LORIA, CNRS & INRIA, France

Stuart Card
User Interface Research Group, Palo Alto Research Center (PARC), USA

John M. Carroll
College of Information Sciences and Technology, The Pennsylvania State University, USA

Jim Foley
Georgia Institute of Technology, USA

Ephraim P. Glinert
National Science Foundation, USA

Vicki L. Hanson
IBM T.J. Watson Research Center, USA

John Karat
IBM T.J. Watson Research Center, USA

Waldemar Karwowski
Center for Industrial Ergonomics, University of Louisville, USA

Sara Kiesler
HCI Institute, Carnegie Mellon University, USA

Arnold Lund
Mobile Platforms Division, Microsoft, USA

Aaron Marcus
Aaron Marcus and Associates, Inc., USA

Dianne Murray
Independent Consultant, UK

Jakob Nielsen
Nielsen Norman Group, USA

Gary M. Olson
School of Information, University of Michigan, USA

Judith S. Olson
School of Information, Ross School of Business, and Department of Psychology, University of Michigan, USA

Sharon Oviatt
Department of Computer Science and Engineering, Oregon Health and Science University, USA

Fabio Paternò
Laboratory on Human Interfaces in Information Systems, ISTI–C.N.R., Italy

Richard Pew
BBN Technologies, USA

Dylan Schmorrow
Office of Naval Research (ONR), USA

Michael Smith
Department of Industrial and Systems Engineering, University of Wisconsin–Madison, USA

Kay Stanney
Industrial Engineering and Management Systems, University of Central Florida, USA

Constantine Stephanidis
Institute of Computer Science, Foundation for Research and Technology-Hellas (ICS-FORTH) Department of Computer Science, University of Crete, Greece

Peter Thomas
Carey Thomas Pty Ltd., Australia

Susan Wiedenbeck
College of Information Science and Technology, Drexel University, USA

Hidekazu Yoshikawa
Department of Socio-Environmental Energy Science, Kyoto University, Japan

SERIES FOREWORD

With the rapid introduction of highly sophisticated computers, (tele)communication, service, and manufacturing systems, a major shift has occurred in the way people use technology and work with it. The objective of this book series on Human Factors and Ergonomics is to provide researchers and practitioners a platform where important issues related to these changes can be discussed, and methods and recommendations can be presented for ensuring that emerging technologies provide increased productivity, quality, satisfaction, safety, and health in the new workplace and the Information Society.

The present volume is published at a very opportune time, when Information Technologies are emerging as a dominant force, both in the workplace, and in everyday life activities. In order for these new technologies to be truly effective, they must provide communication modes and interaction modalities that address the requirements of the user population at large, including individuals with disabilities and older adults, thus making the Information Society universally accessible to the benefit of mankind.

This second edition of the Handbook is published at a time when the field of HCI has experienced a major scientific and technical evolution in the depths and breadths of the subject matter. Currently HCI research and practice is prevalent in over 60 countries and encompasses over 80,000 practitioners and researchers. The subject matter of HCI is published in over 20 journals, over 30 different conferences, and in over 400 books and over 250 company provided services in HCI. There are well over 50,000 individual conference papers, journal articles and technical reports relating to HCI. The second edition of this handbook captures the most important scientific and technical know-how derived from the above sources. Thus, it effective captures the field of HCI. The 67 chapters have been authored by some of the foremost authorities in the field including individuals from academia, industry, and research institutes from around the globe. With over 5,500 references, 400 figures and 100 tables, the handbook provides a wealth of data as well as pointers to the literature for more in depth reading for both researchers and practitioners.

Gavriel Salvendy
Series Editor
Purdue University
and
Tsinghua University
P.R. China

FOREWORD

This handbook celebrates the success of human-computer interaction (HCI). Readers from within the discipline can take satisfaction in their substantial accomplishments: basic research on menus, pointing devices, and screen design as well as design innovations in hot HCI topics such as online communities, information visualization, mobile devices, or interactive games. Practitioners can take pride in their contributions to better products and services, and its growing acceptance by decision-makers.

Newcomers to HCI will find informative section introductions that highlight the key ideas and guide readers to the details. They can ground themselves in the underlying cognitive principles (Part I), appreciate the range of design possibilities (Parts II and III) as well as the diversity of users one must consider (Part V), and recognize the growing sophistication of modern development processes (Part VI).

Outsiders can dip into the topics that concern them now, and expand their reading as their interests grow. They will likely begin with chapters on applications for which they must build or improve user interfaces (Part IV). Visionary readers may find forward-looking ideas to help shape the future, especially the reflections on human values, ethics, and trust (Part VII). Every reader will benefit from the numerous references that connect to the foundational research.

It is satisfying to see the breadth and depth of HCI, and reflect on its future. I kept wondering whether this volume marks this new discipline's emergence from childhood, its youthful adolescence, or young adult status.

The case for childhood is that HCI is just beginning to be accepted in academic departments and corporate boardrooms. Only a few years have passed since the topic became a funding area within governmental research support agencies (e.g. U. S. National Science Foundation and other national counterparts) and the first doctoral programs were announced. On the professional side, usability engineers fight for acceptance, authors propose competing design guidelines, and analysts squabble over appropriate metrics of success.

The argument for adolescence is that the uncertainty of childhood has passed. HCI has already made its mark, as these articles confirm, but this energetic adolescent still has to choose from many possible directions for its future. It could become a basic science like physics and psychology; it could remain an eclectic interdiscipline like bio-physics and socio-linguistics; or it could mature into a professional discipline with scientific foundations like architecture and medicine.

The rationale for the young adult status is that human-computer interaction has already produced interesting theories and stimulated important innovations such as graphical user interfaces, the World Wide Web, email, online communities, instant messaging, and multimedia. It has successfully done its part in bringing information and communications technologies to a diverse set of users and in fashioning devices that are well on their way to universality, ubiquity, and invisibility.

Different readers may come to different conclusions and all be correct. Like any sufficiently large intellectual domain, there is room for many concurrent viewpoints. Those who think HCI is still in childhood can guide its development by engaging in new applications and innovative technologies. They can try to influence HCI's maturation by refining, extending, and applying predictive and explanatory theories, while proposing new theories and integrative frameworks.

Those who see HCI in the adolescent stages can engage in broadening the field by introducing new educational programs, expanding participation in professional societies, and enlarging existing journals. Energetic adolescents should take some risks and explore affiliations with other disciplines such as sociology, semiotics, or nanotechnology. They can expand their experiences by working in other cultures than their own.

Those who see HCI as a young adult can promote its integration into traditional social structures. They can help usability professionals to rise through corporate and government ranks, to be more influential in controlling product development plans, budgets and schedules. They can find the way to engage journalists to tell HCI stories and make HCI issues more visible in related disciplines such as software engineering, networking, or security. HCI advocates still have work to do to gain acceptance by top corporate executives, key government officials, and leading science policy makers.

As for the future, I have high expectations and great hopes for HCI as a mature discipline and profession. We can preserve our youthful infatuation with new and cool technologies, but we must also address adult responsibilities for usable, universal and useful technologies (CHI'99 Research Agenda Workshop):

1. Usable: Current user interfaces and their underlying systems are just too frustrating for most users. A major effort for software developers should be to increase the reliability and usability of every system. The frequency of system reboots, application crashes, and incompatible file formats that stop users

in the middle of their tasks must be reduced. Then the incomprehensible instructions, mysterious menus, and troubling dialog boxes need to be revised to enable users to complete their work promptly and confidently. In our study of frustrations encountered by more than one hundred users, they also made frequent comments about long download times of web pages, dropped sessions for networked applications, and the disruption caused by unsolicited email (spam), and destructive viruses. As the number of users has grown, all these problems have become more serious. Novice users want the benefits of email or web services, but they are the poorly equipped and motivated to overcome the problems that high-tech early adopters proudly conquered. Improved training can help, but improved designs are an important component of making the next generation of interfaces more appealing and the users more successful.

2. Universal: Once systems are made more reliable, then we will be ready to address the challenge of making them universal usable. We must enable people with diverse backgrounds to become successful satisfied users: old & young, men & women, novice & expert, users with varied disabilities, and users with low motivation, low self-confidence, and poor reading skills. We will have to make it possible for users with differing language skills, foreign cultures, and non-traditional social norms to participate in online communities, e-commerce, and e-learning. We must also enable people with varying technologies to benefit from information and communications services; users with slow modems, small screens, or older processors. Advanced technical solutions will be needed to convert web pages to fit small mobile devices and provide web services by cellphone. Finally, we must learn to bridge the gap between what people know and what they need to know. Improved online help, tutorials, email assistance, and customer service are essential. These contributions will often come first from software producers in developed nations, but they need to be pressured to include adequate concerns for the needs of users and opportunities in developing nations.

3. Useful: Usable and universal technologies will make it possible to deliver vital services: life-critical medical systems, military and police protection, safe transportation, and effective job training. Then content providers can create truly useful e-business, e-learning, e-healthcare, e-government, and many other applications. HCI professionals and researchers can bring respect to the discipline by engaging in vital national and international issues such as economic growth, community development, terror prevention, and conflict resolution. By taking responsibility for difficult questions such as the contemporary issues of free speech, intellectual property, voting technologies, pornography control, and privacy we will win respect within and outside our field. We need theories of human needs to understand what is missing and where the greatest opportunities lie. The next major direction I foresee is the enhancement of creativity support tools to make more people more creative more of the time. This ambitious goal can produce a sustainable chain reaction of further creativity and even better tools that more effectively harness each individual's capacity to contribute and participate in shaping the future.

Further signs of adulthood will be the willingness of HCI researchers and professionals to take on the issues that transcend technology. We will gain respect as we contribute to systems that empower individuals, enrich human relationships, encourage cooperation, build trust, and support empathy. Our acceptance of responsibility for the future will be more than a model for other disciplines; it can bring hope for a better life to many people.

Ben Shneiderman

University of Maryland

PREFACE

This second edition of the HCI Handbook provides a fresh perspective on the field of human-computer interaction (HCI). Numerous chapters deal with topics that were not addressed in the first edition. Different authors were recruited to address some topics from the first edition, ensuring that this revision provides new perspectives and insights. Finally, other chapters were updated to ensure that they present the latest in research and practice. While HCI may have emerged from within computing, it was quickly energized by substantial contributions from other fields, such as engineering, psychology, education, and graphic design. The resulting interdisciplinary research has produced important outcomes including an improved understanding of the relationship between people and technology as well as more effective processes for utilizing this knowledge in the design and development of solutions which can increase productivity, quality of life, and competitiveness. HCI now has a home in every application, environment, and device, and is routinely used as a tool for inclusion. HCI is no longer just an area of specialization within more traditional academic disciplines, but has developed such that both undergraduate and graduate degrees are available that focus explicitly on the subject. This handbook should be of tremendous value to practitioners, researchers, students, and academicians regardless of enterprise, because HCI, like computing itself, is now ubiquitous.

Composed of eight sections which include 67 chapters, the nearly 2000 pages of this handbook were authored by over 130 imminent researchers and professionals from around the world. Authors work in academia, industry, and research institutes. The Advisory Board was instrumental throughout the process as they helped shape and refine our vision. We thank the contributing authors and the Advisory Board as their commitment to the success of this volume was unwavering. We are honored to have worked with each one of them. It was also truly a pleasure working with Gavriel Salvendy, Lori Handelman, Anthony Messina, and Anne Duffy whose insights and facilitation were critical to the success of this effort.

The handbook opens with a new perspective on the evolution of HCI, which provides a sense of perspective on the field, recognizes significant visionary developments, and directs the reader to seminal work in the field. Sections covering humans in HCI and computers in HCI follow. These 16 chapters address fundamental issues that affect all design decisions. Next, a collection of 22 chapters spanning two sections address the design of human-computer interactions as well as the design of solutions for specific applications and domains. An entire section titled "Designing for Diversity" is dedicated to addressing the needs of diverse users. This unique collection of 8 chapters sets this volume apart from many other HCI resources. The final 19 chapters explore the state of the art in the development process and the management of HCI. The volume concludes with a chapter that presents interviews with five international experts where they share their perspectives on the past, present, and future of HCI.

Andrew Sears and Julie A. Jacko

September 2006

ABOUT THE EDITORS

Andrew Sears is a Professor of Information Systems and the Chair of the Information Systems Department at UMBC. He is also the director of UMBC's Interactive Systems Research Center. Dr. Sears' research explores issues related to human-centered computing with an emphasis on accessibility. His current projects focus on accessibility, broadly defined, including the needs of individuals with physical disabilities and older users of information technologies as well as mobile computing, speech recognition, and the difficulties information technology users experience as a result of the environment in which they are working or the tasks in which they are engaged. His research projects have been supported by numerous corporations (e.g., IBM Corporation, Intel Corporation, Microsoft Corporation, Motorola), foundations (e.g., the Verizon Foundation), and government agencies (e.g., NASA, the National Institute on Disability and Rehabilitation Research, the National Science Foundation, and the State of Maryland). Dr. Sears is the author or co-author of numerous research publications including journal articles, books, book chapters and conference proceedings. He is Co-Editor-in Chief of the *ACM Transactions on Accessible Computing*. He is also on the editorial board of the *International Journal of Human-Computer Interaction, Universal Access in the Information Society*, and the *Journal of Organizational and End User Computing* and the advisory board of the upcoming *Universal Access Handbook*. He has served on a variety of conference committees including as Conference and Technical Program Co-Chair of the Association for Computing Machinery's Conference on Human Factors in Computing Systems (CHI 2001), Conference Chair of the ACM Conference on Accessible Computing (Assets 2005), and Program Chair for Asset 2004. He is currently Vice Chair of the ACM Special Interest Group on Accessible Computing. He earned his BS in Computer Science from Rensselaer Polytechnic Institute and his Ph.D. in Computer Science with an emphasis on Human-Computer Interaction from the University of Maryland—College Park.

Julie A. Jacko is Professor of Biomedical Engineering, with a joint appointment as Professor in the College of Computing, at the Georgia Institute of Technology and is the author or co-author of over 120 research publications including journal articles, books, book chapters and conference proceedings. She is also the Director of the Center for Interactive Systems within the Health Systems Institute at Georgia Tech. Dr. Jacko's research activities focus on human-computer interaction, human aspects of computing, universal access to electronic information technologies, integrative health, and health care informatics. Her externally funded research has been supported by the Intel Corporation, Microsoft Corporation, the National Science Foundation, NASA, the NIH Agency for Health Care Research and Quality (AHRQ), and the National Institute on Disability and Rehabilitation Research. Dr. Jacko received a National Science Foundation CAREER Award for her research titled, "Universal Access to the Graphical User Interface: Design For The Partially Sighted," and the National Science Foundation's Presidential Early Career Award for Scientists and Engineers, which is the highest honor bestowed on young scientists and engineers by the US government. She is Editor-in-Chief of the *International Journal of Human-Computer Interaction* and she is Associate Editor for the *International Journal of Human Computer Studies*. In 2001 she served as Conference and Technical Program Co-Chair for the ACM Conference on Human Factors in Computing Systems (CHI 2001). She also served as Program Chair for the Fifth ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2002), and as General Conference Chair of ASSETS 2004. In 2006, Dr. Jacko was elected to serve a three-year term as President of SIGCHI. Dr. Jacko routinely provides expert consultancy for organizations and corporations on systems usability and accessibility, emphasizing human aspects of interactive systems design. She earned her Ph.D. in Industrial Engineering from Purdue University.

The Human–Computer Interaction Handbook

Fundamentals, Evolving Technologies, and Emerging Applications

Second Edition

·Introduction·

A MOVING TARGET: THE EVOLUTION OF HCI

Jonathan Grudin
Microsoft Corporation, USA

Preamble: History in a Time of Rapid Obsolescence 2
Definitions: HCI, CHI, Human Factors, Ergonomics, IS, IT 2
Human-Tool Interaction at the Dawn of Computing 2
Three Roles in Early Computing 3
1945–1955: Managing Vacuum Tubes 3
Grace Hopper, Liberating Computer Users 3
1955–1965: Transistors, New Vistas 3
Supporting Operators: The First Systematic HCI Research 3
Visions and Demonstrations 4
Vannevar Bush and the MEMEX 4
J.C.R. Licklider at BBN and ARPA 4
John McCarthy, Christopher Strachey, Wesley Clark . . . 4
Ivan Sutherland and Computer Graphics 5
Douglas Engelbart, Augmenting Human Intellect . . . 5
Ted Nelson's Vision of Interconnectedness 5
Conclusion: Visions, Demos, and Widespread Use 5
1965–1980: HCI Before Personal Computing 6
Human Factors and Ergonomics Embraces Computers . 6
Information Systems 6
Programming: Subject of Study, Source of Change 7
Computer Science: A New Discipline 7
Computer Graphics: realism and interaction 7
Artificial Intelligence: winter follows summer 8
1980–1985: Discretionary Use Comes Into Focus 9
Discretion in Computer Use 9
Minicomputers and Office Automation 10
The Formation of ACM SIGCHI 10
CHI and Human Factors Diverge 12
Workstations and Another AI Summer 12
1985–1995: Graphical User Interfaces Succeed 13
CHI Embraces Computer Science 14
Human Factors and Ergonomics Maintains a Nondiscretionary Use Focus 14
IS Extends Its Range 15
Collaboration Support: OIS Gives Way to CSCW 15
1995–2005: The Internet Era Arrives 16
The Formation of AIS SIGHCI 16
Human Factors and Ergonomics Embraces Cognitive Approaches 16
CHI Evolves, Embraces Design 17
Looking Back: Cultures and Bridges 18
Effects of Varying Discretion 18
Academic, Linguistic, and Generational Cultures 18
Looking Ahead: Trajectories 19
Discretion—Now You See It, Now You Don't 19
Ubiquitous Computing, Invisible HCI? 19
Human Factors and Ergonomics 19
Information Systems 20
Computer-Human Interaction 20
Information Science 20
Conclusion: The Next Generation 20
Appendix: A Few Personal Observations 20
1970: A Change in Plans 20
1973: Three Computing Professions, Only One of Them Hands-on 20
1975: Joining the First Profession of Discretionary Hand-on Users 20
1983: A Chilly Reception for an Early Paper on Discretion in Use 21

1984: First Encounters with IS, Human Factors, and Design 21
1985: The GUI Shock 21
1986: Beyond "The User": Group and Organizational Issues 21
1989: Discovering Contexts of Development, a Major CHI-IS Differentiator 21
1990: Just Words: Terminology Can Matter 21
2006: Reflecting on Bridging Efforts 22
References 22

PREAMBLE: HISTORY IN A TIME OF RAPID OBSOLESCENCE

> "What is a typewriter?" my six-year-old daughter asked.
> I hesitated. "Well, it's like a computer," I began.

A paper widely read 15 years ago concluded with the advice to design a word processor by analogy to something familiar to everyone: a typewriter. Even then, one of my Danish students questioned this reading assignment, noting, "The typewriter is a species on its last legs."

Why study the history of human-computer interaction (HCI)? For most of the computing era, interaction involved 80-column punch cards, paper tape, line editors, 1920-character displays, 1-megabye diskettes, or other extinct species. Are the interaction issues of those times relevant today? No.

Of course, aspects of the human side of HCI change more slowly if at all. Much of what was learned about perceptual, cognitive, social, and emotional processes in interacting with older technologies applies to emerging technologies. However, with human behavior, *what* was learned is often more important than how it came to be learned. Other chapters of the *Handbook* lay out this knowledge.

Nevertheless, in an apparent paradox, the rapid pace of change could make knowledge of the field's history particularly useful for several reasons:

1. Several disciplines are engaged in HCI research and application, but most people are exposed to only one. By seeing how others evolved, we can identify possible benefits and challenges in expanding our focus.
2. The recognition of past visionaries and innovators is part of building a community and inspiring future contributors, even when specific past achievements are difficult to appreciate today.
3. Some visions and prototypes were quickly converted to widespread application, some took decades, and some remain unrealized. By understanding the reasons for different outcomes, we might assess today's visions more realistically.
4. Crystal balls are notoriously unreliable, but anyone planning or managing a career in a rapidly changing field must consider the future, for one thing is certain: It won't resemble the present. Our best chance is to find trajectories that extend from the past through the present.

Accordingly, this account emphasizes disciplinary issues, early visions, and when technologies and practices became widely used. The latter makes this more of a social history than an engineering history that emphasizes "firsts." I point out possible trends and trajectories that you might download into your crystal balls.

A historical account is a perspective. It emphasizes some things, and de-emphasizes or omits others. A history can be wrong in details, but it can never be right in any final sense. Your questions and your interests will determine whether a perspective is useful to you. More information on some topics can be found in prior accounts.

The blueprint for intellectual histories of HCI was established by Ron Baecker in the opening chapters of the 1987 and 1995 editions of *Readings in Human-Computer Interaction*, followed by Richard Pew's wonderfully written chapter in the 2003 version of this *Handbook*. Brian Shackel's (1997) account of European contributions and specialized essays by Brad Myers (1998) on HCI engineering history and Alan Blackwell (2006) on the history of metaphor in design include additional references. Recent years have seen a wave of popular books covering the history of personal computing (e.g. Hiltzik, 1999; Hertzfeld, 2005; Markoff, 2005). This chapter expands on Grudin (2005; 2006). Additional information on some topics can be found in the *Timelines* column in *ACM Interactions* from March 2006.

These writers and I are not trained historians. We lived through much of the computing era as participants and witnesses. Our objectivity can be called into question. I have no personal contributions to defend (self-citations are only to papers that include historical or demographic data), and I conducted many interviews, but we have biases.

Personal experiences can illustrate points and enliven an account by conveying human consequences of changes that otherwise appear abstract or distant. Some readers welcome them, but others find them irritating. I have attempted to satisfy both groups by moving several personal examples to a short Appendix, akin to "deleted scenes" on a DVD.

Definitions: HCI, CHI, Human Factors, Ergonomics, IS, IT

Exploring the HCI literature is complicated by differences in the ways that dozens of simple terms are used. Some of these differences are identified later in this chapter. Here I explain how I use several key terms. HCI is used very broadly to cover work in several disciplines. CHI (computer-human interaction) refers to one narrower focus, associated mainly with computer science, an Association for Computing Machinery Special Interest Group (ACM SIGCHI) and its annual CHI conference. I use human factors (HF), ergonomics (E), and HF&E interchangeably—some writers attach more specialized meanings to ergonomics, which is a term favored in Europe, but there is little agreement among writers and no clear benefit in differentiating them. The Human Factors Society (HFS) became the Human Factors and Ergonomics Society (HFES) in 1992. I use IS (information systems) to refer to the man-

agement discipline that has also been labeled data processing (DP) and management information systems (MIS). The latter was more widely used until recently. I follow common parlance in referring to organizational information systems specialists as "IT professionals (IT pros)." Information systems is to be differentiated from information science, an old field with a new digital incarnation arising in transformed library schools.

HUMAN-TOOL INTERACTION AT THE DAWN OF COMPUTING

A century ago, Frederick Taylor (1911) employed new technologies and methods—moving pictures and statistical analysis—to improve work practices. Time-and-motion studies were most successful with assembly-line manufacturing and other manual tasks. Despite some uneasiness with "Taylorism," as reflected in Charlie Chaplin's popular satire "Modern Times," science and engineering would remain committed to the pursuit of efficiency.

The World Wars accelerated efforts, focused on matching people to jobs, training them, and then designing equipment and jobs to be more easily mastered. Simple flaws in the designs of World War II aircraft controls (Roscoe, 1997) and escape hatches (Dyson, 1979) led to aircraft losses and thousands of casualties. Engineering psychology was born during the war; afterwards, American aviation psychologists created the HFS. Two legacies of the conflict were a greater awareness of the potential of computers and an enduring interest in behavioral requirements for design. For more on this period, see Roscoe (1997) and Meister (1999, 2005).

Early tool use was not discretionary, whether by an assembly-line worker or a pilot. If training was necessary, workers were trained. One research goal was to reduce training time, but much more important was to increase the speed of reliable skilled performance.

Three Roles in Early Computing

Early computer projects employed people in three roles: (a) management, (b) programming, and (c) operation. Managers oversaw design, development, and operation, specifying the programs to be written and distributing the output. A small army of operators was needed. ENIAC, arguably the first general-purpose electronic computer in 1946, was 10 feet tall, covered 1,000 square feet, and consumed as much energy as a small town. Once a program was written, several people loaded it by setting switches, dials, and cable connections. Despite a design innovation that boosted vacuum-tube reliability by operating them at 25% normal power, 50 spent tubes had to be found and replaced on an average day. Large computers failed every several minutes, and some were repaired by people continuously wheeling shopping carts full of vacuum tubes down aisles seems unnecessary.

Computer operation, management, and programming each eventually became a major focus of HCI research. Despite the evolution of computers and our interaction with them, the research spectrum today reflects aspects of this early division of labor.

1945–1955: MANAGING VACUUM TUBES

Reducing operator burden was a key focus of early innovation. Major strides were reducing the time spent replacing or resetting vacuum tubes and the invention of stored-program computers, which could be loaded from tape rather than manually with cables and switches. These endeavors were consistent with the post-war "knobs and dials" human factors or ergonomics approaches. Before long, one computer operator could do work that previously required a team.

Grace Hopper, Liberating Computer Users

As computers became more reliable and capable, programming became a central activity. To improve programmers' interfaces to computers meant to develop languages, compilers, and constructs such as subroutines. Hopper, a pioneer in these areas, described her goal as "freeing mathematicians to do mathematics" (Hopper, 1952; see also Sammet, 1992). These words are echoed in today's usability goal of freeing users to do their work.

1955–1965: TRANSISTORS, NEW VISTAS

Early forecasts that the world would need few computers reflected the limitations of vacuum tubes. More powerful and reliable solid-state computers, first available commercially in 1958, changed everything. As they were acquired to support scientific and engineering tasks outside research settings, less computer-savvy operators needed better interfaces, and people envisioned uses of computers that had been unimaginable for barn-sized machines of limited capability.

Supporting Operators: The First Systematic HCI Research

> "In the beginning, the computer was so costly that it had to be kept gainfully occupied for every second; people were almost slaves to feed it."
>
> —Brian Shackel (1997)

Almost all computer use of this period involved programs and data that were read in from cards or tape. A program then ran without interruption, produced printed, or punched, or tape output, and terminated. This "batch processing" restricted HCI to basic operation, programming, and use of the output, of which only operation involved hands-on computer use.

Low-paid computer operators set switches, pushed buttons, read lights, loaded, and burst printer paper, loaded and unloaded cards, magnetic tapes, and paper tapes, and so on. Teletypes supported direct interaction, commands typed by the operator interleaved with computer responses and status messages. Even-

tually, the paper that scrolled up one line at a time yielded to "glass tty's" (glass teletypes), called visual display units (VDUs) or video display terminals (VDTs), or CRTs for cathode-ray tubes (CRTs), which also scrolled operator commands and computer-generated messages one line at a time. A monochrome terminal restricted to displaying alphanumeric characters cost $50,000 in today's dollars: expensive, but a small fraction of the cost of a computer. A large computer might have one such console, used only by the operator.

Improving the design of buttons, switches, and displays was a natural extension of human factors/ergonomics. Experts in this field authored the first HCI papers. In 1959, British researcher Brian Shackel published the article, "Ergonomics for a Computer," followed in 1962 by "Ergonomics in the Design of a Large Digital Computer Console." These described the redesign of the consoles for the EMIac and EMIdec 2400 analog and digital computers. The latter was the largest computer at the time (Shackel, 1997).

In the United States, the HFS formed in 1957 and focused on improving the efficiency of skilled performance, reducing errors in skilled performance, and training people to achieve skilled performance. Sid Smith's (1963) "Man-Computer Information Transfer" marked the start of a career in the human factors of computing.

Visions and Demonstrations

As transistors replaced vacuum tubes, a wave of imaginative writing, conceptual innovation, and prototype building swept through the research community. Although some of the language now seems dated, notably the use of male generics, many of their key concepts resonate today.

Vannevar Bush and the MEMEX. In one way or another, Vannevar Bush set much of this in motion. He advised U.S. Presidents Franklin Roosevelt and Harry Truman and served as Director of the Office of Scientific Research and Development. His influential 1945 essay, "As We May Think," outlined a multimedia device that he called the MEMEX. Although Bush initially envisioned it as an improbable mechanical device for information storage and retrieval on microfilm, he anticipated many computer capabilities. Bush emphasized the linking of information, which he called associative indexing,

> the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the MEMEX. . . . Any item can be joined into numerous trails. . . . New forms of encyclopedias will appear, ready made with a mesh of associative trails, [which a user could extend].
>
> The lawyer has at his touch the associated opinions and decisions of his whole experience and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest. The physician, puzzled by a patient's reactions, strikes the trail established in studying an earlier similar case and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology. The chemist, struggling with the synthesis of an organic compound, has all the chemical literature before him in his laboratory, with trails following the analogies of compounds and side trails to their physical and chemical behavior.
>
> The historian, with a vast chronological account of a people, parallels it with a skip trail which stops only on the salient items, and can follow at any time contemporary trails which lead him all over civilization at a particular epoch. There is a new profession of trailblazers, those who find delight in the task of establishing useful trails through the enormous mass of the common record.

Sixty years later, professionals make discretionary hands-on use of computers in just these ways. However, 20 years after Bush wrote his essays, hands-on computer use was largely restricted to operators and data-entry personnel who were, in Shackel's words, "almost slaves" tending costly machines.

J.C.R. Licklider at BBN and ARPA. Between 1960 and 1965, there was an outpouring of ideas and systems tied to the newly realized potential of computers. Licklider, a psychologist, played a dual role similar to that of Bush. He wrote influential essays and backed some of the most important research in computer-science history as a manager at Bolt Beranek and Newman (BBN) from 1957–1962 and director of the Information Processing Techniques Office (IPTO) of the Department of Defense Advanced Research Projects Agency (called ARPA and DARPA at different times) from 1962–1964.

BBN conducted extensive computer-related work funded by the government and employed dozens of influential researchers, including John Seely Brown, Richard Pew, and many who also worked at MIT (e.g., John McCarthy, Marvin Minsky, Licklider himself). IPTO funding was crucial in launching computer-science departments and establishing artificial intelligence as a field in the 1960s, in addition to its best-known accomplishment, giving birth to the Internet.

In 1960, Licklider described "man-machine symbiosis," noting, "there are many man-machine systems." He continued, "At present, however, there are no man-computer symbioses—answers are needed." Licklider saw the computer as "a fast information-retrieval and data-processing machine," but he stressed its capability to play a larger role: "One of the main aims of man-computer symbiosis is to bring the computing machine effectively into the formulative parts of technical problems" (pp. 4–5).

This would require more rapid real-time interaction than batch systems of the time supported. In 1962, Licklider and Wes Clark outlined the requirements of a system for "on-line man-computer communication." They identified capabilities that were ripe for development: timesharing of a computer among many users; electronic input-output surfaces for the display and communication of symbolic and pictorial information; interactive, real-time support for programming and information processing; large-scale information storage and retrieval systems; and facilitation of human cooperation. They were also perceptive in recognizing that other desirable technologies, such as speech recognition and natural language understanding, would be very difficult to achieve.

In 1963, Licklider identified his ARPA colleagues as "the members and affiliates of the Intergalactic Computer Network," anticipating the Internet that ARPA would be instrumental in developing (Pew, 2003). His 1965 book *Libraries of the Future* summarized and expanded this vision. Licklider's role in rapidly advancing computer science and HCI is detailed in Waldrop (2001).

John McCarthy, Christopher Strachey, Wesley Clark. McCarthy and Strachey worked out details of timesharing, a crucial step in enabling interactive computing (Fano & Corbato, 1966). Apart from a small number of researchers who could use computers built without concern for cost through military funding, no one could use a computer interactively as long as that required exclusive access. Timesharing allowed several (and later dozens) of simultaneous users at terminals. Languages were developed to facilitate online control and programming of timesharing systems (e.g., JOSS in 1964).

Clark was instrumental in building the TX-0 and TX-2 at MIT's Lincoln Labs to demonstrate timesharing and other innovative concepts. These machines, which cost on the order of $10 million, were factors in establishing the Boston area as a center for computer research (all prices are in 2007 U.S. dollars). A CHI'05 panel discussion of this period that includes Clark and Ivan Sutherland can be viewed online (ePresence, 2006). The TX-2 was the most powerful and capable in the world at the time, but to put it in perspective, it was less powerful than a smartphone is today.

Ivan Sutherland and Computer Graphics. Sutherland's 1963 PhD thesis, describing the Sketchpad system built on the TX-2 with the intention of making computers "more approachable," is arguably the most impressive and influential document in the history of HCI. A nice version edited by Alan Blackwell and Kerry Rodden is available at http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-574.pdf. Sketchpad launched computer graphics, setting into motion a field of research that would have a decisive impact on HCI 20 years later.

Sutherland demonstrated iconic representations of constraints, copying, moving, and deleting of hierarchically organized objects, object-oriented programming concepts, interaction techniques, and approaches to animation. He supported picture construction using a light pen and facilitated visualization by separating the coordinate system used to define a picture from the one used to display it. Sutherland noted the potential for animated graphics in the movie industry. His frank description of what did not work—what he called a "big flop," when engineers found Sketchpad too limited for computer-assisted design (CAD)—enabled others to make rapid progress.

After completing his PhD, Sutherland succeeded Licklider as the director of IPTO in 1964. Among those whose work he funded in this capacity was Douglas Engelbart at the Stanford Research Institute.

Douglas Engelbart, augmenting human intellect. In 1963, Engelbart published "a conceptual framework for the augmentation of man's intellect," and over the next several years he built systems that made great strides toward realizing this vision. He also supported and inspired engineers and programmers who subsequently made major independent contributions.

Echoing Bush and Licklider, Engelbart saw the potential for computers to become congenial tools that people would choose to use interactively. He wrote:

> By "augmenting man's intellect" we mean increasing the capability of a man to approach a complex problem situation, gain comprehension to suit his particular needs, and to derive solutions to the problems. . . . By "complex situations" we include the professional problems of diplomats, executives, social scientists, life scientists, physical scientists, attorneys, designers. . . . We refer to a way of life in an integrated domain where hunches, cut-and-try, intangibles, and the human "feel for a situation" usefully coexist with powerful concepts, streamlined terminology and notation, sophisticated methods, and highly powered electronic aids.

Engelbart used his ARPA funding to develop and integrate an extraordinary set of prototype applications into his NLS system in the span of a few years. In doing so, Engelbart conceptualized and implemented the foundations of word processing, invented or refined input devices including the mouse and multikey control box, and made use of multidisplay environments that integrated text, graphics, and video in windows. In 1968, these would be demonstrated in a sensational 90-minute event at the Fall Joint Computer Conference in San Francisco (http://sloan.stanford.edu/MouseSite/1968Demo.html). The focal point for interactive systems research in the United States appeared to have moved from the East Coast to the West Coast.

Engelbart, an engineer by training, believed in careful human factors testing of systems. He focused on improving efficiency and reducing errors in skilled use, including concern with effects of fatigue and stress. Engelbart also considered training to be a key component, feeling that people should be willing to tackle a difficult interface if it delivered greater power once mastered. The relative importance of optimizing for skilled vs. initial use later became a source of contention and still surfaces in discussions of interaction.

Ted Nelson's vision of interconnectedness. In 1960, while a graduate student in sociology, the inventor of the term "hypertext" founded Project Xanadu to create an easily used computer network. In 1965, he published a paper titled "A file structure for the complex, the changing, and the indeterminate." Nelson continued to produce works (often self-published) with stirring calls for systems to democratize computing through a highly interconnected, extensible network of digital objects (e.g., Nelson, 1973). Xanadu was never fully realized. Nelson did not see the early World Wide Web as an adequate realization of his vision, but features of lightweight technologies such as weblogs, wikis, unstructured collaborative tagging, and search enable many of the activities Nelson envisioned.

Nelson also foresaw the significance of intellectual property issues that would arise in digital domains. Although his solutions were again not fully implemented, they drew attention to the issues.

Conclusion: Visions, Demos, and Widespread Use

Progress in HCI can be understood in terms of the inspiring visions, conceptual advances needed to realize and demo aspects of the visions in working prototypes, and the evolution of widespread practices. Tying these together and enabling the visions to be realized in practice are the remarkable, relentless hardware advances that led to the production of modern personal computers today, which are many millions of times more powerful than the expensive systems used by pioneers.

At the conceptual level, much of the basic foundation for today's graphical user interfaces was in place by 1965, although it

would be a mistake to underestimate the technical invention and detailed work that followed. However, most computer use of that time bore no resemblance to what could be accomplished by personal use of a $10 million custom-built machine. Pew (2003) described the breakthrough 1960 Digital Equipment Corporation (DEC) PDP-1 as "truly a computer with which an individual could interact." The PDP-1 came with CRT display, keyboard, light pen, and paper-tape reader. It cost about $1 million and had the capacity of a Radio Shack TRS 80 20 years later. It required considerable technical and programming support. It was a machine used by a few fortunate computer-savvy researchers.

Licklider's "man-computer symbiosis," Engelbart's "augmenting human intellect," and Nelson's "conceptual framework for man-machine everything" described a world that did not exist, in which attorneys, doctors, chemists, designers, and other professionals were hands-on users of computers out of choice. The reality was that for some time to come, most hands-on use would continue to be routine, nondiscretionary operation. As for the visions, today some of the capabilities are taken for granted, some are just being realized, and others remain elusive.

1965–1980: HCI BEFORE PERSONAL COMPUTING

Control Data Corp. launched the transistor-based 6000 series computers in 1964. In 1965, the first commercial computers based on integrated circuits arrived with the IBM System/360. These powerful systems, later called mainframes to distinguish them from minicomputers, brought computing into the business realm. Each of the three roles in computing—operation, management, programming—became a significant profession.

Operators interacted directly with computers to perform routine maintenance, load and run programs, handle printouts, and so on. As timesharing spread, this hands-on category expanded to include data entry and other repetitive tasks.

Managers oversaw hardware acquisition, software development, operation, routing, and the use of output. They were usually not hands-on users, but people who relied on printed output and reports considered themselves computer users.

Apart from those working in research settings, programmers were rarely direct users until late in this period. Many flowcharted and wrote programs on paper forms. Keypunch operators then punched the program instructions onto cards. These were sent to computer centers for computer operators to run. Printouts and other output were picked up later. Many programmers would use computers directly when they could, but the cost of computer use generally dictated an efficient division of labor.

Human Factors and Ergonomics Embraces Computers

In 1970, Brian Shackel founded the Human Sciences and Advanced Technology (HUSAT) center at Loughborough University in the UK, which was devoted to ergonomics research emphasizing HCI. Sid Smith and other human factors engineers examined a range of input and output issues, notably the representation of information on displays (e.g., S.L. Smith, Farquhar, & Thomas, 1965) but also computer-generated speech (e.g., S.L. Smith & Goodwin, 1970). In 1972, the Computer Systems Technical Group (CSTG) of the HFS formed; soon it was the largest technical group in the society.

Leading publications were the general journal *Human Factors* and, starting in 1969, the computer-focused *International Journal of Man-Machine Studies (IJMMS)*.

The first widely read HCI book was James Martin's 1973 *Design of Man-Computer Dialogues*. A comprehensive survey of interfaces for operation and data entry, it began with an arresting opening chapter that described a world in transition. Extrapolating from declining hardware prices, Martin wrote,

> The terminal or console operator, instead of being a peripheral consideration, will become the tail that wags the whole dog. . . . The computer industry will be forced to become increasingly concerned with the usage of people, rather than with the computer's intestines. (pp. 3–4)

In the mid-1970s, U.S. government agencies responsible for agriculture and social security initiated some large-scale data-processing system development efforts, described in Pew (2003). Although these efforts did not succeed, they led to methodological innovation in the use of style guides, usability labs, prototyping, and task analysis.

In 1980, three significant HF&E books were published: two on VDT design (Cakir, Hart, & Stewart, 1980; Grandjean & Vigliani, 1980) and one of general guidelines (Damodaran, Simpson, & Wilson, 1980). German work on VDT standards, first published in 1981, provided an economic incentive to design for human capabilities by threatening to ban noncompliant products. Later in 1981 a corresponding ANSI standards group formed for office and text systems.

Information Systems

Beginning in 1967, the journal *Management Science* published a column titled "Information Systems in Management Science." Early definitions of IS included "an integrated man/machine system for providing information to support the operation, management, and decision-making functions in an organization" and "the effective design, delivery and use of information systems in organizations" (G. B. Davis, 1974; Keen, 1980; cited in Zhang, Nah, & Preece, 2004). A historical survey of IS research (Banker & Kaufmann, 2004) identified HCI as one of five major research streams. This stream began with a paper on challenges in dealing with computer-generated information (Ackoff, 1967).

Companies acquired expensive business computers to address major organizational concerns. At times, the principal concern was to appear modern (Greenbaum, 1979), but when computers were used, managers could be chained to them almost as tightly as operator and data-entry "slaves." That said, operator or end-user resistance to using a system was a major management concern. The sociotechnical approach to system design was one response; it educated representative workers in technology possibilities and involved them in design in part to increase acceptance of the resulting system (Mumford, 1971).

Cognitive style, a major topic of early IS research, focused on difficulties that managers had communicating with people knowledgeable about computers. IS researchers published HCI articles in management journals and also in the human factors-oriented *IJMMS*. The latter, which became *International Journal of Human-Computer Studies* in 1994, was the 23rd most influential IS journal according to Mylonopoulos and Theoharakis (2001, with their calculation amended in Grudin, 2005, p. 60 note 17).

Programming: Subject of Study, Source of Change

In the 1960s and 1970s, more than 1,000 research papers on variables affecting programming performance were published (Baecker & Buxton, 1987). Most viewed programming in isolation, independent of organizational context. Gerald Weinberg's landmark *The Psychology of Computer Programming* appeared in 1971. In 1980, Ben Shneiderman published *Software Psychology*, and in 1981 Beau Sheil reviewed studies of programming notation (conditionals, control flow, data types), practices (flowcharting, indenting, variable naming, commenting), and tasks (learning, coding, debugging).

Programmers changed their own field through invention. In 1970, Xerox Palo Alto Research Center (PARC) was founded to advance computer technology by developing new hardware, programming languages, and programming environments. It drew researchers and system builders from the labs of Engelbart and Sutherland. In 1971, Allen Newell of Carnegie Mellon University proposed a project to PARC, launched three years later: "Central to the activities of computing—programming, debugging, etc.—are tasks that appear to be within the scope of this emerging theory (a psychology of cognitive behavior)" (quoted in Card & Moran, 1986).

Like HUSAT, also launched in 1970, PARC had a broad research charter. HUSAT focused on ergonomics, anchored in the tradition of nondiscretionary use, one component of which was the human factors of computing. PARC focused on computing, anchored in visions of discretionary use, one component of which was also the human factors of computing. Researchers at PARC and a few other places extended the primarily perceptual-motor focus of human factors to higher-level cognition. HUSAT, influenced by sociotechnical design, extended human factors by considering organizational factors.

Computer Science: A New Discipline

Computer science departments emerged in the mid-1960s. Some arose out of engineering. Computer graphics was one such specialization of particular relevance to HCI; software engineering came later. Other computer science departments originated as applied mathematics; this background was shared by many early artificial intelligence researchers.

Early machines capable of interesting work in these areas were expensive. They were uniformly funded without regard to cost by branches of the military, for which technical success was the sole criterion (Norberg & O'Neill, 1996). ARPA funding under the direction of Licklider, Sutherland, and their successors played a major role. Reliance on massive funding meant that researchers were concentrated at a few centers. These environments bore little resemblance to the batch and timeshared environments of business computing. Hands-on users in research settings were technically savvy. They had less need for low-level interface enhancements.

The computer graphics and AI perspective that developed in these centers differed from that of the HCI researchers of the time, who focused on less expensive, less capable systems that could be studied in many more settings. To HCI, hardware advances meant greater computing capability at a relatively fixed low price. Computer graphics and AI required high levels of computation—hardware advances meant declining cost for a relatively fixed level of computation. Only later would widely available machines be able to support graphical interfaces and AI programming. Nevertheless, between 1965 and 1980 some computer science researchers focused on interaction, which had been part of Ivan Sutherland's initial vision.

Computer Graphics: realism and interaction. In 1968, Sutherland joined David Evans to establish a hugely influential computer graphics lab at the University of Utah. The computer science department there was founded in 1965, one in the first wave that emerged from mathematics and electrical engineering. The western migration continued as students from the lab, including Alan Kay and William Newman (and later Jim Blinn and Jim Clark), went to California. Most graphics systems were built on the DEC PDP-1 and PDP-7. These expensive machines—the list price of a high-resolution display alone was over $100,000 in today's dollars—were capable of multitasking, but graphics programs generally required all of the processing for one task.

In 1973, the Xerox Alto arrived. It was a powerful step toward realizing Alan Kay's vision of computation as a medium for personal computing (Kay & Goldberg, 1977). It was too expensive to be widely used—the Alto never became a product—and not powerful enough to support high-end graphics research. However, it was a machine produced in volume that supported graphical user interfaces of the kind Engelbart had prototyped.

William Newman expressed the result this way: "Everything changed—the computer graphics community got interested in realism, I remained interested in interaction, and I eventually found myself doing HCI" (personal communication). Ron Baecker and Jim Foley were other graphics researchers whose focus shifted to broader interaction issues. Foley and Wallace (1974) identified requirements for designing "interactive graphics systems whose aim is good symbiosis between man and machine," and 18 papers in the first SIGGRAPH conference the same year had "interactive" or "interaction" in their titles.

At Xerox, Larry Tesler and Tim Mott took another step, recognizing that the Alto could support a graphical interface accessible to untrained people. By early 1974, they had developed the GYPSY text editor, which along with Xerox's Bravo editor developed by Charles Simonyi preceded and influenced Microsoft Word (Hiltzik, 1999).

The distinct focus on interaction was given a voice in 1976, when SIGGRAPH sponsored a two-day workshop in Pittsburgh

titled, "User Oriented Design of Interactive Graphics Systems." Participants who were later active in CHI included Jim Foley, William Newman, Ron Baecker, John Bennett, Phyllis Reisner, and Tom Moran.

J.C.R. Licklider and Nicholas Negroponte presented vision papers. UODIGS'76 can be thought of as ending the visionary period, embodying an idea whose time had not quite yet come. Licklider (1976, p. 89) saw it clearly:

> Interactive computer graphics appears likely to be one of the main forces that will bring computers directly into the lives of very large numbers of people during the next two or three decades. Truly user-oriented graphics of sufficient power to be useful to large numbers of people has not been widely affordable, but it will soon become so, and, when it does, the appropriateness and quality of the products offered will to a large extent determine the future of computers as intellectual aids and partners of people.

UODIGS was not repeated and the 150-page proceedings were not cited. Not until 1981 was another such "user-oriented design" conference held, after which such conferences were held every year. Application was not quite at hand; most HCI research remained focused on interaction driven by commands, forms, and full-page menus.

Artificial Intelligence: winter follows summer. In the late 1960s and early 1970s, AI burst onto the scene, promising to transform HCI. It did not go as expected.

Logically, AI and HCI are closely related. What are intelligent machines for if not to interact with people? AI research has influenced HCI: speech recognition and natural language are perennial HCI topics; expert, knowledge-based, adaptive, and mixed-initiative systems have been tried, as have applications of production systems, neural nets, and fuzzy logic; and recently human-robot interaction has attracted attention.

Although some AI features make their way into systems and applications, frequent predictions that more powerful machines would soon bring major AI technologies into widespread use were not borne out. Thus, AI did not come into focus in HCI, and AI researchers have shown limited interest in HCI.

To piece this together requires a brief review of the early history. The term *artificial intelligence* first appeared in a 1955 call by John McCarthy for a meeting on machine intelligence that was held in Dartmouth the next year. Also in 1956, Alan Turing's prescient essay, "Computing Machinery and Intelligence," was reprinted in *The World of Mathematics,* where it attracted attention. (It was first published in 1950, as were Claude Shannon's "Programming a Computer for Playing Chess" and Isaac Asimov's three laws of robotics.) Newell and Simon's logic-theory machine appeared in 1956, after which they focused on developing a general problem solver. McCarthy invented LISP in 1958.

AI pioneers were trained primarily in mathematics and logic, where much can be built from a few axioms and a small set of rules. Mathematics is considered a high form of intelligence, even by nonmathematicians. It is not surprising that AI researchers anticipated that machines that operate logically and tirelessly would make great strides. Nor is it surprising that mathematicians would overlook the complexity and illogic that mark human users and social constructs. Early work on AI focused heavily on theorem proving and on games and problems with a strong logical focus, such as chess and go. In 1988 McCarthy, who espoused predicate calculus as a foundation for AI, summed it up as follows:

> As suggested by the term "artificial intelligence," we weren't considering human behavior except as a clue to possible effective ways of doing tasks. The only participants who studied human behavior were Newell and Simon. (The goal) was to get away from studying human behavior and consider the computer as a tool for solving certain classes of problems. Thus, AI was created as a branch of computer science and not as a branch of psychology.

Strong claims date back to the predawn of AI, when in the summer of 1949 Alan Turing was quoted in the *London Times*:

> I do not see why [the computer] should not enter any one of the fields normally covered by the human intellect, and eventually compete on equal terms. I do not think you can even draw the line about sonnets, though the comparison is perhaps a little bit unfair because a sonnet written by a machine will be better appreciated by another machine.

When he identified speech understanding as important, Licklider realized its difficulty. He predicted that intelligent machines would appear in 10 to 500 years (Pew, 2003). As director of ARPA's Information Processing Techniques Office from 1962–1964, he initiated extensive support for computer science in general and AI in particular. MIT's Project Mac, founded in 1963 by Marvin Minsky and others, initially received $13 million per year, rising to $24 million in 1969. ARPA also sponsored the Artificial Intelligence Laboratory at Stanford Research Institute, AI research at SRI and CMU, and Nicholas Negroponte's Machine Architecture Group at MIT. An early dramatic result, SRI's Shakey the Robot, was featured in 1970 articles in *Life* (Darrach, 1970) and *National Geographic*. Given a simple but nontrivial task, Shakey went to the desired location, scanned and reasoned about the surroundings, and moved objects as needed to accomplish the goal (for Shakey at work: http://www.ai.sri.com/shakey/).

In 1970, Negroponte outlined the case for machine intelligence:

> People generally distrust the concept of machines that approach (and thus why not pass?) our own human intelligence . . . Why ask a machine to learn, to understand, to associate courses with goals, to be self-improving, to be ethical—in short, to be intelligent? . . . Because any design procedure, set of rules, or truism is tenuous, if not subversive, when used out of context or regardless of context. (p. 1)

Negroponte followed this insightful diagnosis with a false inference:

> It follows that a mechanism must recognize and understand the context before carrying out an operation. (p. 1)

Not when the mechanism is guided by a human who is cognizant of the context. However, Negroponte used this to build a case for an ambitious research program:

> Therefore, a machine must be able to discern changes in meaning brought about by changes in context, hence, be intelligent. And to do this, it must have a sophisticated set of sensors, effectors, and processors to view the real world directly and indirectly. . . . A paradigm for fruitful conversations must be machines that can speak and respond to a natural language. . . . But, the tete-à-tete (sic) must be even more direct and fluid; it is gestures, smiles, and frowns that turn a conversa-

tion into a dialogue. . . . Hand-waving often carries as much meaning as text. Manner carries cultural information: the Arabs use their noses, the Japanese nod their heads. . . (pp. 12–13)

Imagine a machine that can follow your design methodology, and at the same time discern and assimilate your conversational idiosyncrasies. This same machine, after observing your behavior, could build a predictive model of your conversational performance. Such a machine could then reinforce the dialogue by using the predictive model to respond to you in a manner that is in rhythm with your personal behavior and conversational idiosyncrasies . . . The dialogue would be so intimate—even exclusive—that only mutual persuasion and compromise would bring about ideas, ideas unrealizable by either conversant alone. No doubt, in such a symbiosis it would not be solely the human designer who would decide when the machine is relevant.

The same year, Negroponte's MIT colleague Minsky took the next step, as reported in *Life*:

In from three to eight years we will have a machine with the general intelligence of an average human being. I mean a machine that will be able to read Shakespeare, grease a car, play office politics, tell a joke, and have a fight. At that point, the machine will begin to educate itself with fantastic speed. In a few months, it will be at genius level and a few months after that its powers will be incalculable. (Darrach, 1970, p. 60)

Other AI researchers told Darrach that Minsky's timetable was somewhat ambitious:

Give us 15 years' was a common remark—but all agreed that there would be such a machine and that it would precipitate the third Industrial Revolution, wipe out war and poverty and roll up centuries of growth in science, education and the arts. (p. 60)

Responding to such calls, ARPA initiated major funding of speech recognition and natural language understanding in 1971. Five years later, disappointed with the progress, ARPA discontinued support for speech and language—for a while.

In Europe, a similar story unfolded. Through the 1960s, AI research expanded in Great Britain, a principal proponent being Turing's former colleague Donald Michie. Then in 1973, the Lighthill Report, commissioned by the Science and Engineering Research Council, reached generally negative conclusions about the prospects for AI systems to scale up to address real-world problems, and almost all government funding was cut off.

The next decade was described as an AI winter.

1980–1985: DISCRETIONARY USE COMES INTO FOCUS

In 1980, most people in HF&E and IS were focused on the down-to-earth business of making efficient use of expensive mainframes. Almost unnoticed was the foreshadowing of a major shift. Less expensive and more capable minicomputers based on LSI technology enabled Digital Equipment Corporation, Wang Laboratories, and Data General to make inroads into the mainframe market. At the low end, home computers gained capability. Growing numbers of student and hobbyist programmers were drawn to these minis and micros, creating a population of hands-on discretionary users.

Then, between 1981 and 1984, the Xerox Star, IBM PC, Apple Lisa, Symbolics and LMI Lisp machines, Sun Microsystems and Silicon Graphics workstations, and the Apple Macintosh were released.

Another major event came on January 1, 1984, when AT&T broke into competing companies. AT&T had had the most employees and the most customers of any U.S. company. Neither customers nor employees had much discretion in technology use, so AT&T and its Bell Laboratories division had focused on improving training and efficiency through Human Factors. In addition to competing in telephony, AT&T entered the PC market in1985 with the ill-fated Unix PC. AT&T and the regional operating companies faced customers who had choices, and their HCI focus broadened accordingly (see Israelski & Lund, 2003).

In general, less-expensive computers created markets for shrinkwrap software, and for the first time, computer and software companies targeted significant numbers of nontechnical hands-on users who would get little or no formal training.

After 20 years, the visions were being realized. Nonprogrammers were choosing to use computers to do their work. The psychology of discretionary users was of particular interest to two groups: (a) psychologists who liked to use computers and (b) technology companies planning to sell to discretionary users. There was a match: Computer and telecommunication companies hired many experimental psychologists. However, before describing this, I will discuss discretionary use and its contrast: the hands-on use required to keep expensive systems running efficiently.

Discretion in Computer Use

Our lives are distributed along a continuum between the assembly-line nightmare of *Modern Times* and utopian visions of completely empowered individuals. To use a technology or not to use it: Sometimes we have a choice, other times we don't. On the phone, I may have to wrestle with speech recognition and routing systems. In contrast, my home computer use is largely discretionary. The workplace often lies in between: Technologies are recommended or prescribed, but we can ignore some injunctions, obtain exceptions, use some features but not others, and join with colleagues to press for changes in policy or availability.

For early computer builders, work was more a calling than a job, but operation required a staff to carry out essential but less interesting tasks. For the first half of the computing era, most hands-on use was by people hired with a mandate. Hardware innovation, more versatile software, and steady progress in understanding the psychology of users and tasks—and transferring that understanding to software developers—led to hands-on users with more choice in what they did and how they did it. Rising expectations played a role—people learned that software is flexible and expected it to be more congenial. Competition among vendors produced alternatives. Today there is more emphasis on marketing to consumers and user friendliness.

Discretion is not all or none. No one *must* use a computer, but many jobs and pastimes require it. True, people can resist, sabotage, use some features but not others, or quit the job. However, a clerk or systems administrator is in a different situation than someone using technology for leisure activity. For an airline-reservation operator, computer use is mandatory. For someone booking his or her own flight, computer use is discretionary. This chapter explores implications of these differences.

Several observers noted the shift toward greater discretion. A quarter century ago, John Bennett (1979) predicted that discretionary use would lead to more concern for usability. A decade later, Liam Bannon (1991) noted broader implications of a shift "from human factors to human actors." However, the trajectory is not always toward choice. Discretion can be curtailed—for example, a word processor is now required for many jobs and is no longer simply an alternative to a typewriter. Even in an era of specialization, customization, and competition, the exercise of choice varies over time and across contexts.

Discretion is one factor of many, but an analysis of its role casts light on the relationships among diverse HCI efforts: why they differ and why they have remained distinct.

Minicomputers and Office Automation

Cabinet-sized computers that could support several people were available from the mid-1960s. Starting with the VAX 11/780, superminis in the late 1970s were capable enough to affect mainframe sales and support integrated suites of productivity tools. In 1980, Digital Equipment Corporation, Data General, and Wang Laboratories were growth companies near Boston. Digital became the second largest computer company in the world.

A minicomputer could handle a database of moderate size or personal productivity tools used from terminals. For "dumb terminals," the central processor handled each keystroke; other minicomputers came with terminals with a processor that supported a person filling out a screen with information that was then sent to the central processor. A mini could support a small group (or "office") with file sharing, applications such as word processing, spreadsheets, and e-mail, and output devices. They were marketed as "office systems," "office automation systems," or "office information systems."

In 1980, the Stanford International Symposium on Office Automation launched a research field that was influential for a decade. Douglas Engelbart contributed two papers to the proceedings (Landau, Bair, & Siegman, 1982). The same year, the American Federation of Information Processing Societies (AFIPS, parent organization of ACM and IEEE at the time) held the first of seven annual Office Automation conferences with an associated product exhibition. Also in 1980, ACM formed a Special Interest Group on Office Automation (SIGOA), which two years later launched the biennial Office Information System Conference. In 1983, *ACM Transactions on Office Information Systems (TOOIS)* emerged, one year after the independent journal *Office: Technology and People.*

Office Information Systems, which focused on the uses of minicomputers at the time, was positioned alongside Management Information Systems (IS), which focused on mainframes. The scope is reflected in the charter of *TOOIS:* database theory, artificial intelligence, behavioral studies, organizational theory, and communications. Minis were accessible database research tools; Digital's PDP series was a favorite for AI researchers; minis were familiar to behavioral researchers who used them to run and analyze experiments, and they became interactive computers of choice for many organizations. Computer-mediated communication was an intriguing new capability that easily supported users at different terminals of the same computer.

The researchers were discretionary computer users, but most office workers did not choose their tools. The term "automation," challenging and exciting to the researchers, conjured up different images for some office workers, some of whom preferred Engelbart's focus on augmentation.

Papers in the SIGOA newsletter, COIS, and *TOOIS* included technical work on database theory, a moderate number of AI papers (the AI winter had not yet ended), decision-support and computer-mediated communication papers from the IS community, and behavioral studies by researchers who later became active in CHI. IS papers were prevalent in the newsletter and technical papers in the journal. The journal was also a major outlet for behavioral studies before *HCI* started in 1985.

Although OA/OIS research was eventually absorbed by other fields, it led the way in addressing a range of important emerging issues, including hypertext, computer-mediated communication, and collaboration support more generally.

The Formation of ACM SIGCHI

Major threads of HCI research are illustrated in Fig 1: Human Factors, Information Systems, and the research focused on discretionary hands-on use that emerged in the 1980s. In 1980, *Human Interaction with Computers,* by Harold Smith and Thomas Green, perched on a cusp. It briefly addressed "the human as a systems component" (the nondiscretionary perspective, p. ix). One third covered research on programming. The remainder addressed "nonspecialist people," discretionary users who were not computer specialists. Smith and Green wrote, "It's not enough just to establish what people can and cannot do; we need to spend just as much effort establishing what people can *and want to do*" (italics in the original; p. viii).

That year, as IBM prepared to launch the PC, a groundswell of attention to computer user behavior was building. IBM had recently added software to hardware as a product focus. Several cognitive psychologists joined an IBM research group that included John Gould, who had engaged in human factors research since the late 1960s. They initiated empirical studies of programming and software design and use. Other psychologists leading recently formed HCI groups included Phil Barnard at the Medical Research Council Applied Psychology Unit (APU) in Cambridge, England; Tom Landauer at Bell Labs; Donald Norman at the University of California, San Diego; and John Whiteside at Digital Equipment Corp.

Xerox PARC and its CMU collaborators were particularly active, continuing work in several areas that proved to have singular influence. The 1981 Star, with its carefully designed graphical user interface, was not a commercial success (nor were a flurry of GUIs that followed, including the Apple Lisa), but it influenced researchers and developers—and of course the Macintosh.

Communications of the ACM created a "Human Aspects of Computing" department in 1980. The next year, Tom Moran edited a special issue of *Computing Surveys* on "The Psychology of the Computer User." Also in 1981, the ACM Special Interest Group on Social and Behavioral Science Computing (SIGSOC) extended its workshop to cover interactive software design and use. In 1982, a conference in Gaithersburg, Maryland, on "Human Factors in Computing Systems" was unex-

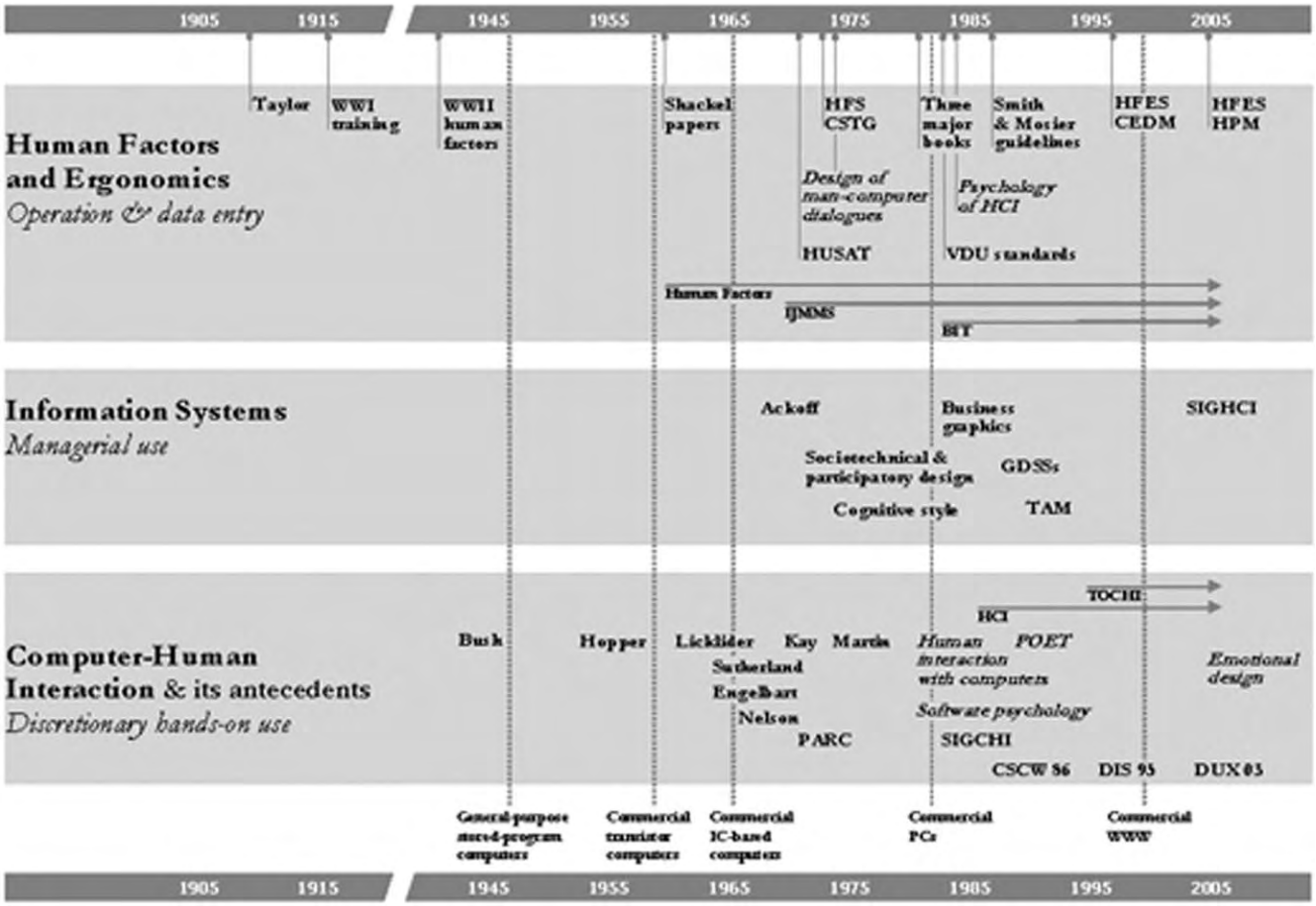


FIGURE 1. HCI events and topics discussed in this chapter. Expansion of acronyms, significance of people and books, and reasons for their placement are in the text.

pectedly well attended. Shortly afterwards, SIGSOC shifted its focus to computer-human interaction and its name to SIGCHI (Borman, 1996).

In 1983, the first CHI conference drew more than 1,000 people. Half of the 58 papers were from the seven research labs just mentioned. Cognitive psychologists in industry dominated the program, although the HFS cosponsored the conference and contributed the program chair Richard Pew, committee members Sid Smith, H. Rudy Ramsay, and Paul Green, and several presenters. Brian Shackel and society president Robert Williges gave tutorials the first day.

The first profession to become discretionary hands-on users was computer programming, as paper coding sheets were discarded in favor of text editing at interactive terminals, PCs, and small minicomputers. Therefore, many early CHI papers, by Ruven Brooks, Bill Curtis, Thomas Green, Ben Shneiderman, and others, continued the psychology-of-programming research thread. IBM Watson researchers also contributed, as noted by John Thomas:

> One of the main themes of the early work was basically that we in IBM were afraid that the market for computing would be limited by the number of people who could program complex systems, so we wanted to find ways for "nonprogrammers" to be able, essentially, to program. (personal communication, October 2003)

Psychologists and studies of editing were so prevalent that in 1984 Thomas Green remarked that, "text editors are the white rats of HCI." As personal computing spread and the same methods were applied to studying other discretionary use, studies of programming gradually disappeared.

CHI focused on novice use. Initial experience is particularly important for discretionary users and the vendors developing software for them. Novice users are also a natural focus when studying new technologies, and a critical focus when more people take up computing each year than did the year before.

Routine, experienced use was still widespread. Computer databases were extensively used by airlines, banks, government agencies, and other organizations. This hands-on activity was rarely discretionary. Managers mainly oversaw development and analyzed data, leaving data entry and information retrieval to people hired for those jobs. Improving skilled data handling was a human factors undertaking. CHI studies of database use were few—I count three over a decade, all focused on novice or casual use.

Fewer European companies produced mass-market software. European research favored in-house development and use. At Loughborough University, HUSAT focused on job design (the division of labor between people and systems) and collaborated with the Institute for Consumer Ergonomics, particularly on product safety. In 1984, Loughborough initiated an HCI graduate program drawing on human factors, industrial engineering, and computer science. The International Conference on HCI (INTERACT) conference, first held in London in 1984 and chaired by Shackel, drew HF&E and CHI researchers.

In his perceptive essay just cited, Bannon urged that more attention be paid to discretionary use while criticizing CHI's heavy emphasis on initial experiences; this may have been a reflection of Bannon's European perspective.

The visionaries were not familiar to many of the CHI researchers who helped realize some of their visions. The 633 references in the 58 papers presented at CHI 83 included many authored by well-known cognitive scientists, but Bush, Sutherland, and Engelbart were not cited at all. Many computer scientists familiar with the early work entered CHI a few years later, and the CHI psychologists eventually discovered and identified with these pioneers who shared their concern for discretionary use, provided conceptual continuity, and bestowed legitimacy on a young enterprise seeking to establish itself academically and professionally.

CHI and Human Factors Diverge

Between 1980 and 1985, researchers at Xerox PARC and CMU introduced another influential research program. Card, Moran, and Newell (1980a, 1980b) described a "keystroke-level model for user performance time with interactive systems" with cognitive components in the goals, operators, methods, selection rules (GOMS) model that were the basis for their landmark 1983 book, *The Psychology of HCI.*

This work was highly respected within CHI although it did not address discretionary, novice use. On the contrary, it focused on the repetitive expert use studied in human factors. In fact, it was explicitly positioned in opposition to the stimulus-response bias of human-factors research:

> Human-factors specialists, ergonomists, and human engineers will find that we have synthesized ideas from modern cognitive psychology and artificial intelligence with the old methods of task analysis. . . . The user is not an operator. He does not operate the computer, he communicates with it. . . . (p. viii)

Newell and Card (1985) noted that human factors had a role in design, but continued,

> Classical human factors . . . has all the earmarks of second-class status. (Our approach) avoids continuation of the classical human-factors role (by transforming) the psychology of the interface into a hard science. (p. 221)

Card wrote:

> Human factors was the discipline we were trying to improve. . . . I personally changed the (CHI conference) call in 1986, so as to emphasize computer science and reduce the emphasis on cognitive science, because I was afraid that it would just become human factors again. (E-mail, June 2004).

> "Hard science, in the form of engineering, drives out soft science, in the form of human factors," wrote Newell and Card. (1985, p. 212)

Ultimately, human-performance modeling drew a modest-but-fervent CHI following. Key goals of the modelers differed from those of practitioners and other researchers. "The central idea behind the model is that the time for an expert to do a task on an interactive system is determined by the time it takes to do the keystrokes," wrote Card, Moran, and Newell (1980b). Modeling was extended to a range of cognitive processes, but remained most useful in helping to design for nondiscretionary users, such as telephone operators engaged in repetitive tasks (e.g., Gray, John, Stuart, Lawrence, & Atwood, 1990). Its role in augmenting human intellect was unclear.

CHI and human factors moved apart, although "Human Factors in Computing Systems" remains the CHI conference subtitle. They were never closely integrated. Most of the cognitive psychologists had turned to HCI after earning their degrees and were unfamiliar with the human-factors research literature. The HFS did not again cosponsor CH, and its researchers disappeared from the CHI program committee. Most CHI researchers who had published in the annual human-factors conference and *Human Factors* journal shifted to CHI, *Communications of the ACM,* and the journal *HCI* established in 1985 by Moran and published by Erlbaum, a publisher of psychology books and journals.

The shift was reflected at IBM T.J. Watson Research Center. John Gould and Clayton Lewis authored a CHI 83 paper that nicely defined the CHI focus on user-centered, iterative design based on prototyping. Watson cognitive scientists helped shape CHI, but Gould's principal focus remained on human factors; he served as HFS president four years later. Symbolically, in 1984, Watson's Human Factors Group faded away and a User Interface Institute emerged.

CHI researchers wanted to be seen as engaged in "hard" science or engineering. The terms *cognitive engineering* and *usability engineering* were adopted. In the first paper presented at CHI 83, "Design Principles for Human-Computer Interfaces," Donald Norman applied engineering techniques to discretionary use, creating "user-satisfaction functions" based on technical parameters. Only years later did CHI loosen its identification with engineering.

Workstations and Another AI Summer

High-end workstations from Apollo, Sun, and Silicon Graphics appeared between 1981 and 1984. Graphics researchers no longer had to congregate in heavily financed labs (notably MIT and Utah in the 1960s; MIT, NYIT, and PARC in the 1970s). Because these workstations did not reach a mass market, graphics research that focused on photorealism and animation did not directly influence HCI more broadly.

The Xerox Star (actually called an Office Workstation), Apple Lisa, and other commercial GUIs appeared, but when the first CHI conference was held in December, 1983, none was succeeding. They were priced too high or ran on processors that were too weak to exploit graphics effectively.

In 1981, Symbolics and LMI introduced workstations optimized to run the Lisp programming language. The timing could not have been more fortuitous. In October of that year, a conference on Next Generation Technology was held in the National Chamber of Commerce auditorium in Tokyo, and, in 1982, the Japanese government announced the establishment of the Institute for New Generation Computer Technology (ICOT) and its 10-year Fifth-Generation project focused on AI. AI researchers in Europe and the United States sounded the alarm. Donald Michie of Edinburgh saw it as a threat to western computer technology, and Ed Feigenbaum of Stanford wrote,

> The Japanese are planning the miracle product. . . . They're going to give the world the next generation—the Fifth Generation—of computers, and those machines are going to be intelligent. . . . We stand, however, before a singularity, an event so unprecedented that predictions are almost silly. . . . Who can say how universal access to machine intelligence—faster, deeper, better than human intelligence—will change science, economics, and warfare, and the whole intellectual and sociological development of mankind? (Feigenbaum & McCorduck, 1983)

At the same time, parallel distributed processing, or neural net models, seized the attention of researchers and media. Used to model signal detection, motor control, semantic processing, and a wide range of phenomena, neural nets represented conceptual and technical advances over earlier AI work on perceptrons, but were of particular interest because the new generation of minicomputers and workstations supported simulation experiments. Production systems, a computer-intensive AI modeling approach with a psychological foundation developed at CMU, gained wider use in research.

These developments triggered an artificial intelligence gold rush. As with actual gold rushes, most of the money was made by those who outfitted and provisioned the prospectors, although generous government funding again flowed to AI researchers. The European ESPRIT and UK Alvey programs invested over $200 million per year starting in 1984 (Oakley, 1990). In the United States, funding for the DARPA Strategic Computing AI program alone, begun in 1983, rose to almost $400 million in 1988 (Norberg & O'Neill, 1996). Investment in AI by 150 U.S. companies was estimated at about $2 billion in 1985 (Kao, 1998).

The unfulfilled promises of the past led to changes this time around. General problem solving was emphasized less, domain-specific problem solving was emphasized more. Terms such as intelligent knowledge-based systems, knowledge engineering, expert systems, machine learning, language understanding, image understanding, neural nets, and robotics were often favored over AI.

In 1983, Raj Reddy of CMU and Victor Zue of MIT criticized the mid-1970s abandonment of funding for speech-processing research, and soon funds were plentiful for that and other AI efforts (Norberg & O'Neill, 1996, p. 238). Johnson (1985) estimated that 800 corporate employees and 400 academics were working on natural language-processing research in1985. Commercial NLU interfaces to databases such as AI Corporation's Intellect and Microrim Clout appeared.

AI optimism is illustrated by two meticulously researched Ovum reports on speech-and language processing (Johnson, 1985; Engelien & McBride, 2001). In 1985, speech-and-language product "revenue" was $75 million, comprised mostly of income from grants and investor capital. Ovum projected that sales would reach $750 million by 1990 and $2.75 billion by 1995. In 1991, sales were under $90 million, but projections were optimistically pegged at $490 million for 1995 and $3.6 billion for 2000.

U.S. corporations banded together to counter the Japanese Fifth-Generation project, jointly funding the Microelectronics and Computer Technology Corporation (MCC). (U.S. antitrust laws were relaxed to allow them to cooperate in this way.) MCC embraced AI, reportedly becoming the leading customer for both Symbolics and LMI. MCC projects included two parallel NLU efforts, work on intelligent advising, and CYC, Douglas Lenat's ambitious project to build an encyclopedic commonsense knowledge base for other programs to consult. In 1984, Lenat predicted that by 1994 CYC would be intelligent enough to educate itself from online texts. Five years later, CYC was reported to be on schedule to "spark a vastly greater renaissance in (machine learning)" (Lenat, 1989).

Knowledge engineering was another facet of AI that involved human interaction, bringing it closer to HCI, at least in theory. The difficulty of eliciting knowledge from experts frustrated researchers whose primary interest was in representing and reasoning about knowledge. This created opportunities for researchers in the HF&E community. European funding directives particularly encouraged work spanning technical and behavioral concerns. *International Journal of Man-Machine Studies* became a major outlet for both HF&E and AI research in the 1980s.

AI interaction with CHI was limited. CHI'83 and CHI'85 sessions covered speech and language, cognitive modeling, knowledge-based help, and knowledge elicitation, but AI technologies did not succeed in the marketplace and were often directed at nondiscretionary users. AI Corporation sold the database interface Intellect primarily to the government before it disappeared. Nor were many AI researchers and developers interested in interaction details: They focused on the power of tools such as EMACS and UNIX, quickly forgetting the painful weeks spent learning needlessly arbitrary commands.

1985–1995: GRAPHICAL USER INTERFACES SUCCEED

> "There will *never* be a mouse at the Ford Motor Company."
> —High-level acquisition manager, 1985

Graphical user interfaces were a disruptive revolution in interaction when they finally succeeded commercially, as were earlier shifts to stored programs and to interaction based on commands, full-screen forms and full-screen menus. Some sectors were affected well before others.

GUIs were particularly attractive to new users. Their success immediately affected the CHI field. However, not until Windows 3.0 succeeded in 1990 did GUIs have much influence among government agencies and business organizations that were the focus of the other HCI researchers. By then the technology was better understood and less disruptive. The early 1990s also saw

the maturation of local area networks and the Internet. This foundation for computer-mediated communication and information sharing was also transformational.

CHI Embraces Computer Science

Apple launched the Macintosh with a 1984 Super Bowl ad pitched at office work, but sales did not follow, and in mid-1985 Apple was in trouble. Then Macs appeared with four times as much RAM, sufficient to manage Aldus PageMaker, Adobe Postscript, and the Apple LaserWriter as they were released, along with Microsoft's Excel and Word for Macintosh. The more powerful Mac Plus arrived in January, 1986. The Mac succeeded where the many commercial GUIs before it had not; it was popular with consumers and the platform for desktop publishing.

Even within CHI, GUIs were initially controversial. They had disadvantages: an extra level of interface code increased development complexity and created reliability challenges; they consumed processor cycles and distanced users from the underlying system, which, many believed, experienced users would have to learn eventually. Carroll and Mazur (1986) showed that GUIs confused and created problems for people familiar with existing interfaces. An influential essay on direct manipulation interfaces, Hutchins, Hollan, and Norman (1986), concluded, "it is too early to tell" how GUIs would fare. GUIs could well prove useful for novices, they wrote, but "we would not be surprised if experts are slower with Direct Manipulation systems than with command language systems." Most HCI research had focused on expert use, so this valid insight seemed significant. However, in a rapidly expanding consumer market, first-time use is critical. Hardware and software advances eliminated other difficulties, and GUIs were around to stay.

The effects within CHI were dramatic. Active topics of research including command naming, text editing, and the psychology of programming were quickly abandoned; more technical topics such as "user-interface management systems" became significant. At a higher level, psychology gave way to computer science as the driving force in interaction design.

Researchers had been engaged in establishing a comprehensive psychological theoretical framework based on formal experiments (Newell & Card, 1985; Carroll & Campbell, 1986; Long, 1989; Barnard, 1991). This was conceivable for constrained command- and form-based interaction, but could not be scaled up to design spaces that included color, sound, animation, and an endless variety of icons, menu designs, window arrangements, and so on. The urgent need was to identify the most pressing problems and find satisfactory rather than optimal solutions. Rigorous experimentation, the principal skill of cognitive psychologists, gave way to faster and less precise assessment methods. Also, to explore the dynamically evolving, unconstrained design space required software engineering expertise.

As a result, the late 1980s saw an influx of computer scientists into the CHI community. The topic HCI became part of the curriculum of many computer-science departments. Computer scientists working on interactive graphics saw CHI as a natural home, as did software engineers interested in interaction, and some AI researchers working on speech recognition, language understanding, and expert systems. Reflecting this shift, in 1994 ACM launched *Transactions on Computer-Human Interaction.* Of course, computer-science researchers were more familiar with early pioneering work than were many of the cognitive scientists who preceded them.

Early PCs and Macs were not easily networked, but as local area networks spread, CHI's focus expanded to include collaboration support. This brought it into contact with efforts in MIS and Office Automation research, which are discussed below.

Human Factors and Ergonomics Maintains a Nondiscretionary Use Focus

HF&E research continued to respond to the needs of government agencies, the military, aviation, and telecommunications. Census, tax, social security, health and welfare, power-plant operation, air-traffic control, ground control for space missions, military logistics, processing text, and voice data for intelligence contribute to the government being the largest consumer of computing.

Most users in these settings are assigned technology. The focus is on skilled use. Small efficiency gains in individual transactions can yield large benefits over time. For routine data entry and other tasks, improvements that may not influence discretionary users can make a difference. After CHI formed, the HFS undertook a study to see how it would affect membership in its Computer Systems Technical Group and found unexpectedly little overlap (Richard Pew, personal communication).

Research funding in HF&E responded to governmental concerns and initiatives. Government also promoted the development of ergonomic standards, in part to help with the problem of defining system requirements for competitive bidding while remaining at arms' length from the potential developers who better understand the technical possibilities. Compliance with standards was specified in a contract.

In 1986, Sid Smith and Jane Mosier published the last in a series of government-sponsored interface guidelines. Nine hundred and forty-four specific design guidelines were organized into sections titled Data Entry, Data Display, Data Transmission, Data Protection, Sequence Control, and User Guidance. They recognized that GUIs would expand the design space beyond the reach of such an already cumbersome, comprehensive set of guidelines that did not cover icons, pull-down or pop-up menus, mice-button assignments, sound, animation, and so on. Requirements definition shifted to specify predefined interface styles and design processes rather than to identify specific features that would be built from scratch.

DARPA's heavily funded Strategic Computing program set out to develop an Autonomous Land Vehicle, a Pilot's Associate, and a Battle Management system. All raised human-factors research issues. These systems would require interactive technologies such as speech recognition, language understanding, and heads-up displays. These might not be used by people who have a choice, but pilots, people guiding autonomous vehicles, and officers under stressful conditions may have no better alternative. Such technologies could also prove useful for professional translators and intelligence analysts, when a phone system provides no alternative, a disability limits keyboard use, or hands are otherwise occupied.

IS Extends Its Range

Although graphical user interfaces were not quickly adopted by organizations, business graphics was important in a research field focused on managerial use. Remus (1984) contrasted tabular and graphic presentations, Benbasat and Dexter (1985) added color as another factor, and many studies followed. The concept of cognitive fit between task and tool was introduced in this context to explain apparently contradictory results in the literature (Vessey & Galletta, 1991). Studies considered online and paper presentation. In practice, color displays were rare in the 1980s; most managers dealt with printed reports.

Involvement of internal end users in the development process was actively discussed (Friedman, 1989). A Scandinavian group set out to empower workers in this way (Bjerknes, Ehn, & Kyng, 1987), but more often, the focus was on increasing user acceptance of the resulting system.

Hands-on managerial use was atypical, but it was central to group-decision support-systems research, which emerged from decision support systems and evolved into group support systems. Computer-supported meeting facility research was conducted in the mid-1980s in several laboratories (e.g., Begeman et al., 1986; DeSanctis & Gallupe, 1987; Dennis, George, Jessup, Nunamaker, & Vogel, 1988). Given the expense of these systems and the decision-maker focus, key research was in schools of management and not in computer-science departments or software companies. Jay Nunamaker's group at the University of Arizona explored approaches to brainstorming, idea organizing, online voting, and other meeting activities. This became a significant IS contribution to research in Computer Supported Cooperative Work (CSCW), discussed next, and also led to the formation of a company to market the technology (Nunamaker, Briggs, Mittleman, Vogel, & Balthazard, 1997).

An influential IS research thread is based on the Technology Acceptance Model (TAM) introduced in F.D. Davis (1989). It focuses on perceived usefulness and perceived usability to improve "white-collar performance" that is "often obstructed by users' unwillingness to accept and use available systems" (p. 319). "An element of uncertainty exists in the minds of decision makers with respect to the successful adoption," wrote Bagozzi, Davis, and Warshaw (1992, p. 664). This managerial view of individual behavior was influenced by Davis's exposure to some early CHI usability research.

High interest in TAM showed that the MIS focus, in which hands-on use was primarily a nondiscretionary operation, data entry, and data retrieval, was shifting as hands-on use spread to white-collar workers who could refuse to play. Contrast IS with CHI: Consumers choose technologies that they perceive to be useful, so CHI assumes perceived utility and rarely considers utility at all. TAM researchers considered utility more important than usability. CHI focused on usability a decade before TAM, albeit more on measures of actual usability than measures of perceived usability. Perception was a secondary "user satisfaction" measure to CHI researchers, who believed (not entirely correctly) that measurable reduction in time, errors, questions, and training would, over time, translate into positive perceptions. "Acceptance" is not in the CHI vocabulary. A discretionary user chooses or adopts, rather than accepts.

Harvard Business Review published "Usability: The new dimension of product design" (March, 1994). In concluding that "user-centered design is still in its infancy," it made no mention of CHI. The communities remained largely isolated.

Collaboration Support: OIS Gives Way to CSCW

In the late 1980s, three research communities focused on small-group communication and information sharing. The Office Automation/Office Information System field was already there. Declining costs of computing attracted MIS researchers focused on organizational decision making to group decision making more generally, as previously noted. The proliferation of LAN networks encouraged some CHI researchers to move from individual productivity to a quest for "killer apps" that would appeal to groups.

Although the OA/OIS field had led the way, it declined and largely disappeared in this period. The minicomputer platform for much of the work did not survive the competition from PCs and workstations. The concept of "office" or group was problematic: organizations and individuals are relatively sharply defined persistent entities with goals and needs, but small groups often have ambiguous memberships and processes that shift when one member joins or departs. People who work together often fall under different budgets, making technology acquisition complicated when it is not organization wide. Conference series and journals dropped the identification with offices.

First automation lost favor as a term; ACM SIGOA shifted to SIGOIS (Office Information Systems) in 1986, the same year the annual AFIPS OA conferences ended. By 1991, the term "Office" itself began to disappear: *Transactions on Office Information Systems* became *Transactions on Information Systems; Office: Information and People* became *Information Technology and People*; Conference on Office Information Systems became Conference on Organizational Communication Systems.

The AI summer that was one component of the OA/OIS effort ended as AI failed to meet expectations: massive funding did not deliver a Pilot's Associate, Autonomous Land Vehicle, or Battle Management system for the military of automated offices for enterprises. CHI conference sessions on language processing diminished early, but sessions on modeling, adaptive interfaces, advising systems, and other uses of intelligence in interfaces increased through the late 1980s before declining in the 1990s. AI research did not disappear, but funding became scarce, employment opportunities dried up, and conference participation dropped off.

Building on a 1984 workshop (Greif, 1985), the 1986 Computer Supported Cooperative Work conference brought together researchers from diverse disciplines interested in issues of communication, information sharing, and coordination. Participants came primarily from IS, OIS, CHI, distributed AI, and anthropology. Four of 13 CSCW program committee members and many papers were from schools of management, with similar numbers from OIS.

A field seemed to coalesce in 1988, with the publication of *Computer-Supported Cooperative Work,* edited by Irene Greif, and SIGCHI's management of the biennial CSCW conferences

held in North America. However, representation from IS and OIS faded. Scandinavian cooperative or participatory design researchers remained active longer. Although they shared the IS focus on organizational development and use rather than commercial software development, they also shared CHI's focus on discretionary use, albeit with a different slant: empowering workers to control the design of their workplaces. Ethnographers, whose ethic is also to act to benefit their informants, in this case technology users rather than technology managers, rose in influence in CSCW.

Some IS researchers shifted participation to a series of less selective annual Groupware conferences that started in 1992. They also created a newsletter: *Groupware Report* listed many relevant conferences, but not CSCW. Eventually, many in IS settled on the Collaboration Technology track of the annual HICSS conference as a prejournal publication arena, with some participating in the Organizational Computing Systems (1991–1995) and GROUP (1997–present) descendants of the Conference on Office Information Systems.

The cultural issues that pulled apart the groups initially active in CSCW are addressed in the Discussion section. However, CSCW remains a strong research area. It has attracted a broad swath of HCI researchers long enough to foster mutual understanding, ranging from highly technical work on group undo to thick ethnographies of workplace activity, from studies of IM dyads to scientific collaboratories involving hundreds of people over several continents and many years. For details see the *Handbook* chapter "Groupware and Computer Supported Cooperative Work" by prominent CSCW researchers Gary and Judy Olson.

1995–2005: THE INTERNET ERA ARRIVES

How did the spread of the Internet and emergence of the Web affect the different HCI research threads? CHI researchers were relatively Internet savvy. Although excited by the prospects, they took this change in stride. Over time, though, the nature of research, development, and use were affected. The change was not disruptive to human factors and ergonomics, either. The Web was initially a return to a relatively familiar form-driven interface style, and it was not a locus of routine work for many people. However, these developments had a seismic impact on IS, so that is where this section begins.

The Formation of AIS SIGHCI

Computer users in organizations were no longer "almost slaves" devoted to maximizing computer use—screen savers vied with solitaire to be the main consumer of processor cycles. Embrace of the Internet created more porous organizational boundaries. Employees download free software such as instant-messaging clients, music players, and weblog tools inside the firewall despite IT concerns about productivity and security. These are not the high-overhead applications of the past. Another change over time is that home use of software has reduced employee patience with poor interactive software at work. In addition,managers who were hands-off users in the 1980s became late adopters in the 1990s and are now hands-on early adopters of technologies that benefit them.

These changes affect use, but the Web had a more dramatic effect on IS research. Corporate IT departments had focused on internal operations, but suddenly organizations were creating Web interfaces to vendors and customers. The Internet bubble revealed how little was understood in these areas, but neither online presence and services nor business-to-business systems disappeared when the bubble burst. Portals proliferated: the Web became an essential business tool. IT professionals tasked with providing interfaces to highly discretionary external customers found themselves in much the same place CHI had been 20 years earlier, whether they realized it or (most often) not.

In 2001, the Association for Information Systems (AIS) established the Special Interest Group in Human-Computer Interaction (SIGHCI). The founders defined HCI by citing 12 works by CHI researchers (Zhang, et al., 2004, p. 148) and made it a priority to bridge to CHI and the Information Science community (Zhang, 2004, p. 2). SIGHCI's broad charter includes a range of organizational issues, but published work focuses on interface design for e-commerce, online shopping, online behavior "especially in the Internet era," and effects of Web-based interfaces on attitudes and perceptions. Eight of the first 10 papers in SIGHCI-sponsored journal issues cover Internet and Web behavior.

Human Factors and Ergonomics Embraces Cognitive Approaches

In 1996, the Human Factors and Ergonomics Society formed a new technical group, Cognitive Engineering and Decision Making. It became the largest technical group in the society. A decade earlier, this would have seemed an unlikely development: Senior human factors researchers disliked cognitive approaches, and it was in the CHI field that cognitive engineering was being used in this sense (Norman, 1982; 1986).

Even more astonishing would have been the fact that in 2005, human performance modeling would be a new, thriving technical group in HFES. The technical group was started by Wayne Gray and Dick Pew, both of whom participated in CHI in the 1980s. Human performance modeling was the Card, Moran, and Newell (1983) effort to reform the discipline of human factors from the outside. Work had continued, focused on expert performance (e.g., a special issue of *HCI,* Vol. 12, Number 4, 1997). Today the reform effort is appropriately positioned within human factors, focused largely on nondiscretionary use. See Michael Byrne's chapter on cognitive architectures for more on this topic.

HF&E has to a large degree shaped government funding. The U.S. National Science Foundation Interactive Systems program—subsequently renamed Human-Computer Interaction—was described as follows:

> The Interactive Systems Program considers scientific and engineering research oriented toward the enhancement of human-computer com-

munications and interactions in all modalities. These modalities include speech/language, sound, images and, in general, any single or multiple, sequential, or concurrent, human-computer input, output, or action." (National Science Foundation, 1993)

One NSF HCI program manager reported that his proudest accomplishment was doubling the already ample funding for natural language understanding. NSF established a separate Human Language and Communication Program in 2003, but speech-and-language research continued to draw heavy funding support in the HCI and Accessibility programs, and lighter support in AI and other NSF programs.

Two subsequent NSF HCI program managers chose to emphasize "direct-brain interfaces" or "brain-computer interaction," using brainwaves, implants, or other means. These program managers rarely attended CHI conferences, where one finds little work on speech, language, or direct-brain interaction. Whether or not these technologies prove useful, they will not appear soon in many homes or offices. A review committee in 2003 noted that a random sample of NSF HCI grants included none by prominent CHI researchers (National Science Foundation, 2003).

Human-factors research on computer use in general has dispersed. Within HFES, the Computer Systems Technical Group has declined in membership, but HCI issues now appear in most branches of human factors, from telecommunications to medical systems.

CHI Evolves, Embraces Design

The steady flow of new hardware, software features, applications, and systems insures that initial and early use of digital technology is always present, is important to technology producers, and raises new research issues. CHI has tracked this flow of innovation, generally picking up an innovation at the point it attracts a wide audience.

As an application matures, use may become routine. Many people now have little choice about using e-mail and word processing, for example. Such technologies get less attention as CHI directs its gaze at discretionary use of the moment: instant messaging, weblogs, collaboration technology, Web design, ubiquitous computing, mobile computing, and social computing, and so on. New technologies have raised new issues, such as privacy, and encouraged new methods, such as ethnography. At a more abstract level, there is continuity at CHI: there is continued exploration of input devices, communication channels, information-visualization techniques, and design methods.

The growing participation in an Internet that seemed to become more reliable and support higher bandwidth every month through the mid-1990s increased the focus on the use of real-time communication technologies, and quasi-real-time technologies such as e-mail. The Web temporarily stopped much of this, however, by shifting attention to less indirect interaction via static sites.

The Web is like a new land mass, a new continent. First came the explorers, posting flags here and there. Then came the first attempts at settlement, in the form of virtual worlds research and development. Few of these first pioneers survived. There was little to do in virtual worlds, except for multiplayer games and simulations. However, in recent years, people, especially young people, are shifting major portions of their work and play online, including reliance on online reference, digital photos, social software, preferences for digital documents and online shopping, and multiplayer games. This is reflected in CHI research topics, although game design has not yet attracted the attention it merits.

The Web has curtailed some research focused on self-contained personal productivity tools. Despite high development and maintenance costs, representing knowledge in application software was appealing when external information resources were limited. Now that so much information is available online, including the ability to locate and access knowledgeable people, static knowledge representation is less useful. In contrast, adaptive systems that build and maintain local knowledge can play a greater role. Steady progress in machine learning is influencing productivity tools—although hyperbolic forecasts have not disappeared.

The psychologists and computer scientists who formed CHI considered interface design to be a scientific and engineering undertaking. Their focus on performance assumed that people eventually choose the most efficient alternatives. Because human discretion involves aesthetic preferences and invites marketing and nonrational persuasion, as computing costs came down this view could not be sustained. However, it held on longer in CHI than in SIGGRAPH, where aesthetic appeal motivated much of the research. Even now, the study of enjoyment in CHI is labeled "funology" (Blythe, Monk, Overbeeke, & Wright, 2003) lest someone think that the researchers are too relaxed about it.

Visual designers participated in graphical interface research from early on. Aaron Marcus began working full-time on computer graphics in the late 1960s. William Bowman's 1968 book *Graphic Communication* was a strong influence on the Xerox Star, for which the designer Norm Cox's icons were chosen in an evaluation described in Bewley et al., 1983). However, graphic designers were usually seen as secondary (Evenson, 2005). In 1995, building on working-group meetings at recent conferences, SIGCHI initiated Designing Interactive Systems (DIS), a biennial conference drawing some visual designers and many systems designers. In 2003, SIGCHI, SIGGRAPH, and the American Institute of Graphic Arts (AIGA) initiated the Designing for User Experience (DUX) conference series that fully embraces visual and commercial design.

The evolution of CHI is reflected in the influential contributions of Donald Norman. A cognitive scientist who introduced the term *cognitive engineering*, he presented the first CHI 83 paper. It defined "User Satisfaction Functions" based on speed of use, ease of learning, required knowledge, and errors. His influential 1988 book *Psychology of Everyday Things (POET)* focused on pragmatic usability. Its 1990 reissue as *Design of Everyday Things* reflected a field refocusing on invention. Fourteen years later, he published *Emotional Design: Why We Love (or Hate) Everyday Things*, stressing the role of aesthetics in our response to objects.

Design's first cousin, marketing, has been poorly regarded in the CHI community (see Marcus, 2004). However, website design forces the issue. Site owners often wish to keep users on a

site; whereas users may prefer to escape quickly. Consider supermarkets, where items that most shoppers want are positioned far apart, forcing people to traverse aisles so other products can beckon. CHI professionals usually align themselves with "end users," but when designing for a site owner, they face a stakeholder conflict. This was not true in the past: Designers of individual productivity tools had negligible conflict of interest with prospective customers. Marketing is likely to find a place in CHI, perhaps as "brandology."

LOOKING BACK: CULTURES AND BRIDGES

Despite a significant common focus and a dynamic environment with shifting alliances, the three major threads of HCI research—human factors and ergonomics (HF&E), Information Systems (IS), and Computer-Human Interaction (CHI)—have not merged. They have not even interacted as much as one might expect—and not for lack of trying. The HFS was a 50% cosponsor of the first CHI conference. CSCW set out to bring together CHI and IS, and AIS SIGHCI has engaged in another such effort. Understanding the obstacles to interaction provides insight into the nature of these disciplines.

Effects of Varying Discretion

A principal distinction is that HF&E and IS arose before discretionary hands-on use was widespread, whereas CHI made that its focus. Researchers in the first two examined organizational as well as technical issues, the latter came only slowly to consider organizational context. Researchers in HF&E and IS shared journals; for example, Benbasat and Dexter (1985) was published in *Management Science* and cited five human factors articles.

The difference in user motivation affected methods. The psychologists who shaped HF&E and CHI were trained to test hypotheses about behavior in laboratory experiments. Experimental subjects agree to follow instructions for an extrinsic reward. This is a good model for nondiscretionary use, but not for discretionary use. CHI researchers relabeled them "participants," which sounds volitional, but discovered that lab findings require confirmation in real-world settings more often than is true for ergonomics studies.

Traditional ergonomic goals apply—fewer errors, faster performance, quicker learning, greater memorability, and being enjoyable—but the emphasis differs. For power-plant operation, error reduction is key and performance enhancement is good. Other goals are less critical. In other settings, a formal experiment is necessary to show that a new interface will shave a few seconds from a repetitive operation.

In contrast, consumers often respond to visceral appeal and initial experience at the expense of long-term usability and utility. In assessing designs for mass markets where catching obvious problems is more significant than striving for an optimal solution, less-rigorous studies (e.g., discount usability, Nielsen, 1989) are adequate and time-consuming qualitative approaches (e.g., personas, Pruitt, and Adlin, 2006) and can provide a deeper understanding of users that can be drawn upon as new possibilities arise. Unlike HF&E, CHI slowly abandoned its roots in scientific theory and engineering. This did not impress rigorously experimental HF&E or theory-oriented IS researchers. The controversial psychological method of verbal reports, developed by Newell and Simon (1972), was applied to design as the thinking-aloud method by Clayton Lewis (1983; Lewis & Mack, 1982). Perhaps the most widely used CHI method, it led some researchers in the other areas to characterize CHI people as wanting to talk about their experiences rather than doing research.

Academic, Linguistic, and Generational Cultures

HF&E and IS also share the traditional academic culture of the sciences: Conferences are venues for work in progress, and journals are repositories for polished work. In contrast, for CHI and other U.S. computer-science disciplines, conference proceedings are the final destination for most work. Journals are secondary. Outside the United States, computer science retains more of a journal focus, perhaps due to the absence of professional societies that archive proceedings. This circumstance impedes communication across disciplines and continents. Researchers in journal cultures chafe at CHI's rejection rates; CHI researchers are dismayed by the relatively unpolished work at other conferences.

Accepting only around of 20% of submissions, CHI conferences are selective. With few exceptions, HF&E and IS conferences have acceptance rates of 50% or greater. On the other hand, CHI journals receive fewer submissions and have higher acceptance rates. (See Grudin, 2005.) Many CHI researchers report that journals are not relevant, and I estimate that as little as 10% of work in CHI-sponsored conferences reaches journal publication. In contrast, an IS track organizer for Hawaii International Conference on System Sciences 2004 estimated that 80% of research there progressed to a journal (Jay Nunamaker, HICCS-38 presentation, January 2004).

A linguistic divide also set CHI apart. HF&E and IS used the term *operator*; in IS, *user* could be a manager who used printed computer output, not a hands-on end user. Within CHI, *operator* was demeaning, *user* was always hands-on, and *end user* seemed a superfluous affectation. In HF&E and IS, *task analysis* generally refers to an organizational decomposition of work, or, especially more recently, and a broader analysis that includes external factors; in CHI it was a cognitive decomposition, such as breaking a text editing move operation into select, cut, select, paste. In IS, *implementation* meant deployment of a system in an organization; in CHI it was a synonym for development. *System*, *application*, and *evaluation* also had markedly different connotations or denotations. Significant misunderstandings and rejections resulted from failure to recognize these distinctions.

Different perspectives and priorities were reflected in attitudes toward standards. Many HF&E researchers contributed to the development of standards, believing that standards contribute to efficiency and innovation. A widespread CHI view was that standards inhibit innovation. There are elements of truth in both views, and positions may have converged as Internet and

Web standards were tackled. However, the attitudes reflected the different demands of government contracting and commercial software development. Specifying adherence to standards is a useful tool for those preparing requests for proposals, but compliance with standards can make it more difficult for a product to differentiate itself.

A generational divide also existed. Many CHI researchers grew up in the 1960s and 1970s, and did not appreciate the HF&E orientation toward military and government systems, or the inability (still occasionally observed) of HF&E and IS "man-machine interaction" researchers to adopt gender-neutral terminology. It added to the difficulty of generating enthusiasm for building bridges and exploring literatures.

LOOKING AHEAD: TRAJECTORIES

The future of HCI will be tremendously varied, dynamic, and full of surprises. The intent in this chapter is to provide a perspective or framework with which to view other chapters or topics of interest and assess how they might develop. In this section, I extrapolate from some observations about the past and present state of the field.

Discretion—Now You See It, Now You Don't

We exercise choice more at home than at work, a lot when buying online, none when confronted by a telephone answering system, considerable when young and healthy, less when constrained by injury or aging. Software that was discretionary yesterday is indispensable today. The need to collaborate forces us to adopt common systems and conventions.

Consider a hypothetical team. In 1987, one member still used a typewriter, and others chose different word processors. All exchanged printed documents. One emphasized by underlining, another by *italicizing*, a third by **bolding.** In 1997, group members wanted to share documents digitally, so they had to adopt the same word processor and conventions. Choice was curtailed; it had to be exercised collectively. Today, it suffices to share documents in PDF format, so in 2007 the team is using different word processors again. Tomorrow perhaps I could personalize my view to see in italics what you see as bold.

Shackel (1997) noted this progression under the heading "From Systems Design to Interface Usability and Back Again." Early designers focused at the system level; operators had to cope. When the PC merged the roles of operator, output user, and program provider, the focus shifted to the human interface and choice. Then individual users again became components in fully networked organizational systems. When a technology becomes mission-critical, as e-mail did for many in the 1990s, discretion is gone.

The converse also occurs. Discretion increases when employees download free software and demand capabilities they have at home. Managers are less likely to mandate the use of a technology that they use and find burdensome. Even in the military, where language-processing systems appealed to military officers, the situation changed when they became hands-on users:

> Our military users . . . generally flatly refuse to use any system that requires speech recognition. . . . Over and over and over again, we were told 'If we have to use speech, we will not take it. I don't even want to waste my time talking to you if it requires speech. . . .' I have seen generals come out of using, trying to use one of the speech-enabled systems looking really whipped. One really sad puppy, he said 'OK, what's your system like, do I have to use speech?' He looked at me plaintively. And when I said 'No,' his face lit up, and he got so happy." (Forbus, 2003; see also Forbus, Usher, & Chapman, et al., 2003)

As familiar applications become essential and security concerns curtail openness, one might expect discretion to recede. However, Moore's law, competition, and the phenomenal ease of sharing bits seem to guarantee that a steady flow of unproven technologies will find their way to us.

Ubiquitous Computing, Invisible HCI?

Norman (1988) wrote of "the invisible computer of the future." Like motors, he speculated, computers would be present everywhere and visible nowhere. We interact with clocks, refrigerators, and cars. Each has a motor, but there is no human-motor interaction specialization. A decade later, at the height of the Y2K crisis and the Internet bubble, computers were more visible than ever. We may always want a multipurpose display or two, but part of Norman's vision is materializing. However, with computers embedded everywhere, concern with interaction is also everywhere.

Perhaps HCI, too, will become invisible through omnipresence. Interaction with digital technology is becoming part of everyone's research and the three major HCI fields are losing participation.

Human Factors and Ergonomics. David Meister, author of The History of Human Factors and Ergonomics, stresses the continuity of HF&E in the face of technology change:

> Outside of a few significant events, like the organization of HFS in 1957 or the publication of Proceedings of the annual meetings in 1972, there are no seminal occurrences . . . no sharp discontinuities that are memorable. A scientific discipline like HF has only an intellectual history; one would hope to find major paradigm changes in orientation toward our human performance phenomena, but there is none, largely because the emergence of HF did not involve major changes from pre-World War II applied psychology. In an intellectual history, one has to look for major changes in thinking, and I have not been able to discover any in HF. (e-mail to author, 7 September 2004)

Membership in the Computer Systems Technical Group of HFES has declined sharply, but technology use is stressed in many technical groups, such as Cognitive Engineering and Decision Making, Communication, Human Performance Modeling, Internet, System Development, and Virtual Environment. Nor can Aging, Medical Systems, or other technical groups avoid "invisible computers."

Information Systems. As IS thrived during the Y2K crisis and Internet bubble years, other management-school disciplines—finance, marketing, operations research, organizational behavior—become more technically savvy. When the bubble burst and enrollments declined, IS was left with a less well-defined niche. IS research issues, including HCI, remain significant, but this cuts two ways. With the standardization and outsourcing of IT functions, Web portals and business-to-business ties get more attention. These bring in economic and marketing considerations, making it easier to outsource HCI functions to the traditional management disciplines.

Computer-Human Interaction. This nomadic group started in psychology and obtained a place at the table in computer science—in some departments. Lacking a well-defined academic niche, CHI's identity is tied to its conference, where participation peaked in 2001. Conferences that are more specialized thrive. As technologies appear and attract a critical mass at an ever-increasing pace, researchers may start new conferences or just blog their findings. Soon after the Web came into view, annual WWW conferences drew papers on HCI issues. Now we see conferences on ubiquitous and pervasive computing, agents, design, emerging technologies, and so on. HCI is invisibly present in each. High computer science conference rejection rates and a new generational divide could accelerate this dispersion of effort.

Information Science

In 2005, the first annual "i-Conference" drew deans and faculty from information schools to State College, Pennsylvania. Nineteen universities with information programs are listed on the Web site http://iconference.ist.psu.edu/content/view/23/37/. These schools vary considerably: Some are transformed library schools, some have closer ties to information systems, and others formed relatively recently, focusing on digital libraries and other issues. Together they represent an interesting development, in that they have drawn HCI researchers from HF&E, IS, and CHI. Information science also has conceptual and human links to the office information system thread, which had information retrieval and language-processing elements that have been part of the digital library research.

Within these schools, the issues of academic cultures are being addressed on almost a daily basis. It is too early to say how this will evolve, but information science will be a significant player in HCI. Design and information are two foci active in 2007: Design is a compensation for past neglect, and information is a new world opening before us.

CONCLUSION: THE NEXT GENERATION

Moore's law ensures that landscapes will continue to shift, providing new forms of interaction to explore and new practices to improve. The first generation of computer researchers, designers, and users grew up without computers. Then came a generation who began using computers as students; they entered workplaces and changed the way technology was used. Now a generation has grown up with game consoles and cell phones, absorbing an aesthetic of technology design, communicating with IM and text messaging, developing skills at searching, browsing, tagging, synthesizing, and acquiring multimedia authoring talent via digital cameras and blogs. They are entering workplaces, and they will change everything once again.

However it comes to be defined and wherever it is studied, HCI is still in its early days.

APPENDIX: A FEW PERSONAL OBSERVATIONS

My career from 1973 to 1993 followed a common path: computer programmer, cognitive psychology student, who worked first in HCI R&D industry, and then academia. My experiences were not special; I was one of many in a position to see and feel events and changes. I describe some of them here to add texture and a sense of human impact. My interest in history arose from the feeling of being disrupted and swept along by unexpected forces. My first try at understanding was called, "The Computer Reaches Out" (Grudin, 1990)—I saw computers slowly reaching into the world and changing us in ways we did not foresee.

1970: A Change in Plans

I am a student, reading and believing the *Life* magazine article describing super-human intelligent computers arriving in several years. I conclude that if we make it through the next few years, we can count on machines to do all useful work. Humans should focus on what they enjoy. I shifted from physics to mathematics, from politics to literature.

1973: Three Computing Professions, Only One of Them Hands-On

Looking for my first job in 1973, I found three computer job categories in the *Boston Globe* classifieds: operators, programmers, and systems analysts. Not qualified to be a highly paid analyst, I interviewed for some low-paid, hands-on operator jobs, and landed a programming job with Wang Laboratories, which was then a small electronics company. In two years at that job, I never saw the computer my programs ran on. We flowcharted on paper and coded on coding sheets, which a secretary sent to be punched and verified. A van carried the stack of cards 20 miles to a computer center, and later that day or the next I got the printout (e.g., "Error in Line 20").

1975: Joining the First Profession of Discretionary Hand-On Users

In 1975, the company acquired a few teletype terminals with access to the WYLBUR line editor, developed at the Stanford

Linear Accelerator. Some programmers chose to abandon paper and became hands-on computer users.

1983: A Chilly Reception for an Early Paper on Discretion in Use

My first HCI publication showed that people may choose a slower interface for aesthetic or other reasons even when familiar with a more efficient alternative (Grudin & MacLean, 1984). A senior colleague asked us not to publish it. He worked on improving expert performance efficiency through cognitive human performance modeling. He felt that a demonstration that greater efficiency could be undesirable would be a distraction, saying, "Sometimes the larger enterprise is more important than a small study."

1984: First Encounters with IS, Human Factors, and Design

I returned to Wang Laboratories, now a large minicomputer company. I was influenced there by another cognitive psychologist, Susan Ehrlich. She was in a marketing research group and later managed the human factors group. She introduced me to IS literature, which I found difficult to understand. I attended Boston-area chapter meetings of both the HFS and SIGCHI. I noted the cultural differences but felt CHI could learn from human factors. I decided to label myself human factors engineer, a conscious-if-futile gesture to counter CHI antipathy toward human factors. I drove to Cambridge to see the newly released Macintosh. Few software engineers had the visual design skills that I realized would become important, so at work I found industrial designers of the "boxes" who could be attracted to software interface design.

1985: The GUI Shock

In the early 1980s, I was one of many cognitive psychologists working on command naming. This was an important application in the era of command-line interfaces, but our greater ambition was to develop a comprehensive theoretical foundation for HCI. The success of the Mac in 1985 curtailed interest in command names. For most of us, it also dashed the hope of developing a comprehensive psychology of HCI. No one would build on our past work—a depressing thought. We had to choose: Am I a cognitive psychologist or a computer professional?

1986: Beyond "The User": Group and Organizational Issues

I joined MCC, an industry research consortium. Between jobs I worked on two papers, each addressing a different major challenge encountered in product development. From 1984 to 1986, I had worked on several products or features intended to support groups rather than individual users. These had not done well. Why was group support so challenging? In addition, it was painfully evident that organizational structures and development processes were badly suited to interactive software development. What could be done about it? These two issues formed the basis of much of my research for the next decade and made CSCW my natural home.

1989: Discovering Contexts of Development, a Major CHI-IS Differentiator

I was invited to spend two years at Aarhus University in Denmark. Within weeks of arriving in a country with little commercial software development, I recognized the significance of the differences in the conditions that govern product, in-house, and contract development of interactive software, differences that shape CHI, IS, and software engineering perspectives. Sorting this out led to my first library research into historical information (Grudin, 1991). Consulting journals and magazines long untouched in dusty library corridors felt like wandering through an archaeological site.

1990: Just Words: Terminology Can Matter

I felt a premonition in 1987 when my IS-oriented colleague Susan Ehrlich titled a paper "Successful Implementation of Office Communication Systems." By *implementation,* she meant introduction into organizations. To me, *implementation* was a synonym for coding or development. Sure enough, the ACM editor asked her to change *implementation* to *adoption* (Ehrlich, 1987). What she called *systems,* I called *applications.* It left me uneasy. Would my dear friend Language get in the way?

In 1990, I set out to teach HCI at Aarhus, describing the focus as "user-interface evaluation." My new colleagues seemed embarrassed. Weeks later, a book written by one of them was published (Bødker, 1990). Its first sentence was a quotation: "Design is where the action is, not evaluation." Now I was embarrassed. In their in-house development world, with its dogma of getting the design right up front, development projects often took 10 years or more. Evaluation occurred at the end, when only cosmetic changes were possible—thus the negative stigma. In commercial product development, evaluation of the previous version, competitive products, and (ideally) prototypes was integral to design. Evaluation is central to iterative design. It was also the skill experimental psychologists brought to the table. We considered it a *good* thing.

Later in 1990, I participated in a panel on task analysis at a European conference. To my dismay, this IS-oriented group defined *task analysis* differently than I did. To them, it meant an organizational task analysis: tasks as components in a broad work process. In CHI, it meant a cognitive task analysis: breaking a simple task into components; for example, is "move text" better thought of as select-delete-paste or select-move-place?

Also in 1990, en route to give a job talk at UC Irvine, my first lecture to an IS audience at the UCLA Anderson School of Management ended badly when the department head asked a question. It seemed meaningless, so I replied cautiously. He rephrased the question. I rephrased my response. He started again, then stopped and shrugged as if to say, "this fellow is hopeless." When I saw him a few months later, he was visibly astonished

to learn that his Irvine friends were hiring me. Later, I understood the basis of our failure to communicate: We attached different meanings to the word *users*. In CHI, it means hands-on computer users. IS *users* often never used a keyboard—they identified software requirements, managed development, read printed output and reports, and so on. His question had focused on users who were not hands on. To me, all use was hands on, so the question made no sense.

A book could be written about the word *user*. From a CHI perspective, the IS *user* was called "customer." Consultants use *client*. In IS, the hands-on user was the end user. In CHI's parlance, *end user* and *user* were one and the same—a person who both entered data and used the output—so end user seemed a superfluous or odd affectation. Human factors used operator, which CHI considered demeaning. In software engineering, user typically denoted a tool user, namely a software engineer.

I usually consider words a necessary but uninteresting medium for conveying meaning, but these experiences led to an essay on unintended consequences of language (Grudin, 1993).

2007: Reflecting on Bridging Efforts

I have been a minor participant in efforts to find synergies between CHI and human factors, office information systems, information systems (in both CSCW and AIS SIGHCI), and Design. A "sixties person," I experienced generational and cultural divides. Some of us avoided publishing in "man-machine" conferences and journals, and many of my MCC colleagues joined the consortium to avoid "Star Wars" military projects. We lived through disputes between cognitive psychologists and radical behaviorist or strictly perceptual-motor psychologists. Many CHI researchers shifted from journals to conferences as preferred publication venues, and from hypothesis-driven research to build-and-assess research.

Some differences fade over time, but terminology continues to impede communication. Conference reviewers are often irritated by acronyms used by authors from other fields. Writing a chapter for an IS-oriented book, my coauthor and I wrangled at great length with the editor over terminology (Palen & Grudin, 2002).

A final example: In this research, I reviewed the literature on TAM, the model of White-collar employee perceptions of technology that is heavily cited in IS but never in CHI. I repeatedly had difficulty using a search engine to return to TAM references. On the third such occasion, I saw why: TAM stands for "Technology Acceptance Model," but I always typed in "Technology Adoption Model." Nondiscretionary acceptance vs. discretionary adoption: different biases led to different terminology, and to error and confusion.

References

Ackoff, R. L. (1967). Management misinformation systems. *Management Science, 14*, B147–B156.

Asimov, I. (1950). *I, robot.* New York: Gnome Press.

Baecker, R., & Buxton, W. (1987). A historical and intellectual perspective. In R. Baecker & W. Buxton, *Readings in HCI: A multidisciplinary approach* (pp. 41–54). San Francisco: Morgan Kaufmann.

Baecker, R., Grudin, J., Buxton, W., & Greenberg, S. (1995). A historical and intellectual perspective. In R. Baecker, J. Grudin, W. Buxton, & S. Greenberg, *Readings in HCI: Toward the Year 2000* (pp. 35–47). San Francisco: Morgan Kaufmann.

Bagozzi, R. P., Davis, F. D., & Warshaw, P. R. (1992). Development and test of a theory of technological learning and usage. *Human Relations, 45*(7), 660–686.

Banker, R. D., & Kaufmann, R. J. (2004). The evolution of research on Information Systems: A fiftieth-year survey of the literature in Management Science. *Management Science, 50*(3), 281–298.

Bannon, L. (1991). From human factors to human actors: The role of psychology and HCI studies in system design. In J. Greenbaum & M. Kyng (Eds.), *Design at Work* (pp. 25–44). Hillsdale, NJ: Erlbaum.

Barnard, P. (1991). Bridging between basic theories and the artifacts of HCI. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 103–127). Cambridge: Cambridge University Press.

Begeman, M., Cook, P., Ellis, C., Graf, M., Rein, G., & Smith, T. (1986). Project Nick: Meetings augmentation and analysis. *Proceedings Computer-Supported Cooperative Work 1986,* 1–6.

Benbasat, I. & Dexter A. S. (1985). An experimental evaluation of graphical and color-enhanced information presentation. *Management science, 31*(11), 1348–1364.

Bennett, J. L. (1979). The commercial impact of usability in interactive systems. In B. Shackel (Ed.), *Man-computer communication* (Vol. 2.). Maidenhead: Infotech State-of-the-Art, Pergamon-Infotech.

Bewley, W. L., Roberts, T. L., Schroit, D., & Verplank, W. L. (1983). Human factors testing in the design of Xerox's 8010 "Star" office workstation. *Proceedings CHI'83* (pp. 72–77). New York: ACM.

Bjerknes, G., Ehn, P., & Kyng, M. (Eds.). (1987). *Computers and Democracy—a Scandinavian Challenge.* Aldershot, UK: Avebury.

Blackwell, A. (2006). The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction, 13,* 4, 490–530.

Blythe, M. A., Monk, A. F., Overbeeke, K. & Wright, P. C. (Eds.). (2003). *Funology: From usability to user enjoyment.* New York: Kluwer.

Borman, L. (1996). SIGCHI: the early years. *SIGCHI Bulletin, 28*(1), 1–33.

Bowman, William J. (1968). *Graphic Communication.* New York: John Wiley.

Bush, V. (1945). As we may think. *The Atlantic Monthly, 176,* 101–108.

Bødker, S. (1990). *Through the Interface: A Human Activity Approach to User Interface Design.* Mahwah, NJ: Lawrence Erlbaum Associates.

Cakir, A., Hart, D. J. & T. F. M. Stewart, T. F. M. (1980). *Visual display terminals.* New York: Wiley.

Card, S. K., & Moran, T. P. (1986). User technology: From pointing to pondering. *Proceedings of the Conference on the History of Personal Workstations* (pp. 183–198). New York: ACM.

Card, S. K., Moran, T. P., & Newell, A. (1980a). Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology, 12*, 396–410.

Card, S. K., Moran, T. P., & Newell, A. (1980b). Keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396–410.

Card, S., Moran, T. P., & Newell, A. (1983). *The Psychology of HCI.* Mahwah, NJ: Lawrence Erlbaum Associates.

Carroll, J. M., & Campbell, R. L. (1986). Softening up hard science: Response to Newell and Card. *HCI, 2*(3), 227–249.

Carroll, J. M., & Mazur, S. A. (1986). Lisa learning. *IEEE Computer, 19*(11), 35–49.

Damodaran, L., Simpson, A., & Wilson, P. (1980). *Designing systems for people.* Manchester, UK: NCC Publications.

Darrach, B. (1970, November 20). Meet Shaky: The first electronic person. *Life Magazine, 69*(21), 58B–68.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–339.

Davis, G. B. (1974). *Management information systems: Conceptual foundations, structure, and development.* New York: McGraw-Hill.

Dennis, A., George, J., Jessup, L., Nunamaker, J., & Vogel, D. (1988). Information technology to support electronic meetings. *MIS Quarterly, 12*(4), 591–624.

DeSanctis, G., & Gallupe, R. B. (1987). A foundation for the study of group decision support systems. *Management Science, 33*, 589–610.

Dyson, F. (1979). *Disturbing the universe.* New York: Harper & Row.

Ehrlich, S. F. (1987). Strategies for encouraging successful adoption of office communication systems. *ACM Transactions on Office Information Systems, 5*(4), 340–357.

Engelbart, D. (1963). A conceptual framework for the augmentation of man's intellect. In P. Howerton & D. Weeks (Eds.), *Vistas in information handling* (Vol. 1, pp. 1–29). Washington, DC: Spartan Books.

Engelien, B., & McBryde, R. (1991*). Natural language markets: Commercial strategies.* London: Ovum Ltd.

ePresence (2006). Early interactive graphics at MIT Lincoln Labs. Retrieved March 13, 2007 from http://epresence.tv/archives/ or http://www.billbuxton.com/Lincoln.html

Evenson, S. (2005, February). *Design and HCI highlights.* Presented at the HCIC 2005 Conference, Winter Park, Colorado.

Fano, R., & Corbato, F. (1966). Timesharing on computers. *Scientific American 214*(9), 129–140.

Feigenbaum, E. A., & McCorduck, P. (1983). *The Fifth Generation: Artificial Intelligence and Japan's computer challenge to the world.* Reading, MA: Addison-Wesley.

Foley, J. D., & Wallace, V. L. (1974). The art of natural graphic man-machine conversation. *Proceedings of the IEEE, 62*(4), 462–471.

Forbus, K. (2003, May). Sketching for knowledge capture. [Lecture.] Unpublished raw data.

Forbus, K. D., Usher, J., & Chapman, V. (2003). Qualitative spatial reasoning about sketch maps. *Proceedings of the Innovative Applications of AI* (pp.). Menlo Park: AAAI.

Friedman, A. (1989). *Computer systems development: History, organization and implementation.* New York: Wiley.

Gould, J.D., & Lewis, C. (1983). Designing for usability—Key principles and what designers think. *Proceedings of CHI'83* (pp. 50–53). New York: ACM.

Grandjean, E., & Vigliani, A. (1980). Ergonomics aspects of visual display terminals. London: Taylor and Francis.

Gray, W. D., John, B. E., Stuart, R., Lawrence, D., & Atwood, M. E. (1990). GOMS meets the phone company: Analytic modeling applied to real-world problems. *Proceedings of Interact'90* (pp. 29–34). Amsterdam: North Holland.

Greenbaum, J. (1979). *In the name of efficiency.* Philadelphia: Temple University.

Greif, I. (1985). Computer-Supported Cooperative Groups: What are the issues? *Proceedings AFIPS Office Automation Conference* (pp.). Montvale, NJ: AFIPS Press.

Greif, I. (ed.) (1988). *Computer-Supported Cooperative Work: A book of readings.* San Mateo, CA: Morgan Kaufmann.

Grudin, J. (1990). The computer reaches out: The historical continuity of interface design. *Proceedings of CHI'90* (pp. 261–268). NY: ACM.

Grudin (1991). Interactive systems: Bridging the gaps between developers and users. *IEEE Computer, 24*(4), 59–69.

Grudin, J. (1993). Interface: An evolving concept. *Communications of the ACM, 36*(4), 110–119

Grudin, J. (2005). Three faces of HCI. *IEEE Annals of the History of Computing, 27*(4), 46–62.

Grudin, J. (2006). Human Factors, CHI, and MIS. In P. Zhang & D. Galletta (eds.), *HCI in MIS (I): Foundations.* Armonk, NY: M. E. Sharpe.

Grudin, J., & MacLean, A. (1984). Adapting a psychophysical method to measure performance and preference tradeoffs in HCI. *Proceedings of INTERACT'84* (pp. 338–342). Amsterdam: North Holland.

Hertzfeld, A. (2005). *Revolution in the valley: The insanely great story of how the Mac was made.* Sebastopol, CA: O'Reilly Media.

Hiltzik, M. A. (1999). *Dealers of lightning: Xerox PARC and the dawn of the computer age*. New York: HarperCollins.

Hopper, G. (1952). The education of a computer. *Proceedings of ACM Conference*, reprinted in *Annals of the History of Computing, 9*(3–4), 271–281, 1987.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design* (pp. 87–124). Mahwah, NJ: Lawrence Erlbaum Associates.

Israelski, E., & Lund, A. M. (2003). The evolution of HCI during the telecommunications revolution. In J. A. Jacko & A. Sears (Eds.), *The HCI handbook.* Mahwah, NJ: Erlbaum.

Johnson, T. (1985). *Natural language computing: The commercial applications.* London: Ovum Ltd.

Kao, E. (1998). The history of AI. Retrieved March 13, 2007, from http://www.generation5.org/content/1999/aihistory.asp

Kay, A., & Goldberg, A. (1977). Personal dynamic media. *IEEE Computer 10*(3), 31–42.

Keen, P. G. W. (1980). MIS research: reference disciplines and a cumulative tradition. In *First International Conference on Information Systems* (pp. 9–18). Chicago: Society for Management Information Systems.

Landau, R., Bair, J. & Siegmna, J. (Eds.). (1982, March). Emerging office systems. Extended proceedings of the Stanford International Symposium on Office Automation, Norwood, NJ.

Lenat, D. (1989). When will machines learn? *Machine Learning, 4*, 255–257.

Lewis, C. (1983). The 'thinking aloud' method in interface evaluation. Tutorial given at CHI'83.

Lewis, C., & Mack, R. (1982). Learning to use a text processing system: Evidence from "thinking aloud" protocols. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 387–392). New York: ACM.

Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transactions of Human Factors in Electronics HFE-1*, 1, 4–11.

Licklider, J. C. R. (1965). *Libraries of the future.* Cambridge, MA: MIT Press.

Licklider, J. C. R. (1976). User-oriented interactive computer graphics. In *Proc. SIGGRAPH workshop on User-oriented design of interactive graphics systems,* 89–96. New York: ACM.

Licklider, J. C. R, & Clark, W. (1962). On-line man-computer communication. *AFIPS Conference Proceedings, 21*, 113–128.

Lighthill, J. (1973). Artificial intelligence: A general survey. In J. Lighthill, N. S. Sutherland, R. M. Needham, H. C. Longuet-Higgins, & D. Michie (Eds.), *Artificial intelligence: A paper symposium*. London: Science Research Council of Great Britain.

Long, J. (1989). Cognitive ergonomics and HCI. In J. Long & A. Whitefield (Eds.), *Cognitive ergonomics and HCI* (pp. 4–34). Cambridge: Cambridge University Press.

March, A. (1994). Usability: the new dimension of product design. *Harvard Business Review, 72*(5), 144–149.

Marcus, A. (2004). Branding 101. *ACM Interactions, 11*(5), 14–21.
Markoff, J. (2005). *What the dormouse said: How the 60s counter-culture shaped the personal computer.* London: Viking.
Martin, J. (1973). *Design of man-computer dialogues.* New York: Prentice-Hall.
McCarthy, J. (1960). Functions of symbolic expressions and their computation by machine, part 1. *Comm. ACM, 3*(4), 184–195.
McCarthy, J. (1988). B. P. Bloomfield, *The question of artificial intelligence: Philosophical and sociological perspectives*. *Annals of the History of Computing, 10*(3), 224–229.
Meister, D. (1999). *The history of human factors and ergonomics.* Mahwah, NJ: Lawrence Erlbaum Associates.
Meister D. (2005). HFES history. *HFES 2005–2006 directory and yearbook* (pp. 2–3). Santa Monica: Human Factors and Ergonomic Society.
Mumford, E. (1971). A comprehensive method for handling the human problems of computer introduction. *IFIP Congress, 2*, 918–923.
Myers, B. A. (1998). A brief history of HCI technology. *ACM interactions, 5*(2), 44–54.
Mylonopoulos, N. A., & Theoharakis, V. (2001). Global perceptions of IS journals. *Comm. ACM, 44*(9), 29–32.
National Geographic. (1970, November). Behold the computer revolution.
National Science Foundation. (1993). *NSF 93–2: Interactive Systems Program Description*. 13 January 1993.
National Science Foundation. (2003). *NSF Committee of Visitors Report:* Information and Intelligent Systems Division. 28 July 2003.
Negroponte, N. (1970). *The architecture machine: Towards a more humane environment.* Cambridge: MIT Press.
Nelson, T. (1968). A file structure for the complex, the changing, and the indeterminate. *Proceedings of the ACM National Conference* (pp. 84–100). New York: ACM.
Nelson, T. (1973). A conceptual framework for man-machine everything. *Proceedings of the National Computer Conference* (pp. M21–M26). Montvale, New Jersey: AFIPS Press.
Newell, A., & Card, S. K. (1985). The prospects for psychological science in HCI. *HCI, 1*(3), 209–242.
Newell, A., & Simon, H. A. (1956). The logic theory machine: A complex information processing system. *IRE transactions on information theory* IT-2, 61–79.
Newell, A., & Simon, H. A. (1972). *Human problem solving.* New York: Prentice-Hall.
Nielsen, J. (1989). Usability engineering at a discount. In G. Salvendy & M.J. Smith (Eds.), *Designing and using human-computer interfaces and knowledge based systems* (pp. 394–401). Amsterdam: Elsevier.
Norberg, A. L., & O'Neill, J. E. (1996). *Transforming computer technology: Information processing for the Pentagon* (pp. 1962–1986). Baltimore: Johns Hopkins.
Norman, D. A. (1982). Steps toward a cognitive engineering: Design rules based on analyses of human error. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 378–382). New York: ACM.
Norman, D. A. (1983). Design principles for human-computer interfaces. *Proceedings of CHI'83* (pp. 1–10). New York: ACM.
Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User centered system design* (pp. 31–61). Mahwah, NJ: Lawrence Erlbaum Associates.
Norman, D. A. (1988). *Psychology of everyday things.* New York: Basic Books.
Norman, D. A. (2004). *Emotional Design: Why We Love (or Hate) Everyday Things.* New York: Basic Books.
Nunamaker, J., Briggs, R. O., Mittleman, D. D., Vogel, D. R. & Balthazard, P. A. (1997). Lessons from a dozen years of group support systems research: A discussion of lab and field findings. *Journal of Management Information Systems, 13*(3), 163–207.
Oakley, B. W. (1990). Intelligent knowledge-based systems—AI in the U.K. In R. Kurzweil (Ed.), *The age of intelligent machines* (pp. 346–349). Cambridge, MA: MIT Press.
Pew, R. (2003). Evolution of HCI: From MEMEX to Bluetooth and beyond. In J. A. Jacko & A. Sears (Eds.), *The HCI handbook* (pp. 1–17). Mahwah, NJ: Lawrence Erlbaum Associates.
Polen, L., & Grudin, J. (2002). Discretionary adoption of group support software. In B. E. Munkvald, *Implementing collaboration technologies in industry,* (pp. 159–180). London: Springer-Verlag.
Proceedings of the joint conference on easier and more productive use of computer systems. (1981). New York: ACM. (See http://portal.acm.org)
Pruitt, J., & Adlin, T. (2005). *The Persona lifecycle: Keeping people in mind throughout product design.* San Francisco: Morgan Kaufmann.
Remus, W. (1984). An empirical evaluation of the impact of graphical and tabular presentations on decision-making. *Management Science, 30*(5), 533–542.
Roscoe, S. N. (1997). *The adolescence of engineering psychology.* Santa Monica, CA: Human Factors and Ergonomics Society.
Sammet, J. (1992). Farewell to Grace Hopper—End of an era! *Comm. ACM, 35*(4), 128–131.
Shackel, B. (1959). Ergonomics for a computer. *Design, 120*, 36–39.
Shackel, B. (1962). Ergonomics in the design of a large digital computer console. *Ergonomics, 5*, 229–241.
Shackel, B. (1997). HCI: Whence and whither? *Journal of ASIS, 48*(11), 970–986.
Shannon, C. E. (1950). Programming a computer for playing chess. *Philosophical magazine, 7*(41), 256–275.
Sheil, B. A. (1981). The psychological study of programming. *ACM Computing Surveys, 13*(1), 101–120.
Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems.* Cambridge, MA: Winthrop.
Smith, H. T., & Green, T. R. G. (1980). *Human interaction with computers.* Orlando, FL: Academic.
Smith, S. L. (1963). Man-computer information transfer. In J. H. Howard (Ed.), *Electronic information display systems* (pp. 284–299). Washington, DC: Spartan Books.
Smith, S. L., Farquhar, B. B., & Thomas, D. W. (1965). Color coding in formatted displays. *Journal of Applied Psychology, 49*, 393–398.
Smith, S. L., & Goodwin, N. C. (1970). Computer-generated speech and man-computer interaction. *Human Factors, 12*, 215–223.
Smith, S. L., & Mosier, J. N. (1986). *Guidelines for designing user interface software* (ESD-TR-86-278). Bedford, MA: MITRE.
Sutherland, I. (1963). Sketchpad: A man-machine graphical communication system. Unpublished doctoral dissertation, MIT.
Taylor, F. W. (1911). *The principles of scientific management*. New York: Harper.
Turing, A. (1949, June 11). Letter in *London Times*. See Highlights from the Computer Museum report Vol. 20, Summer/Fall 1987. http://edthelen.org/comp-hist/TheCompMusRep/TCMR-V20.html
Turing, A. (1950). Computing machinery and intelligence. *Mind,* 49, 433–460. Republished as "Can a machine think?" in J. R. Newman (Ed.), *The world of mathematics,* (Vol. 4, pp. 2099–2123). New York: Simon & Schuster.
Vessey, I., & Galletta, D. (1991). Cognitive fit: An empirical test of information acquisition. *Information Systems Research, 2*(1), 63–84.
Waldrop, M. M. (2001). *The dream machine: J.C.R. Licklider and the revolution that made computing personal.* New York: Viking.
Weinberg, G. (1971). *The psychology of computer programming.* New York: Van Nostrand Reinhold.
Zhang, P. (2004). AIS SIGHCI three-year report. *SIGHCI newsletter, 3*(1), 2–6.
Zhang, P., Nah, F. F.-H., & Preece, J. (2004). HCI studies in management information systems. *Behaviour & Information Technology, 23*(3), 147–151.

Part

I

HUMANS IN HCI

◆ 1 ◆

PERCEPTUAL-MOTOR INTERACTION: SOME IMPLICATIONS FOR HCI

Timothy N. *Welsh*
University of Calgary

Daniel J. *Weeks*
Simon Fraser University

Romeo Chua
University of British Columbia

David Goodman
Simon Fraser University

Perceptual-Motor Interaction:
A Behavioral Emphasis **28**
Human Information Processing and Perceptual-Motor Behavior 28
Translation, Coding, and Mapping 29
Perceptual-Motor Interaction:
Attention and Performance **30**
Attention 30
Characteristics of Attention 30
Shifts and Coordinate Systems of Attention 30
Stimulus Characteristics and Shifts of Attention 31
Facilitation and inhibition of return 32
Action-Centered Attention 33
The model of response activation 34
Attention and action requirements 35
Attention and stimulus-response compatibility 36
Perceptual-Motor Interaction in Applied Tasks:
A Few Examples **37**
Remote Operation and Endoscopic Surgery 37
Personal Digital Assistants 37
Eye-Gaze vs. Manual Mouse Interactions 38
Summary **38**
Acknowledgments **38**
References **39**

PERCEPTUAL-MOTOR INTERACTION: A BEHAVIORAL EMPHASIS

Many of us can still remember purchasing our first computers to be used for research purposes. The primary attributes of these new tools were their utilities in solving relatively complex mathematical problems and performing computer-based experiments. However, it was not long after that word processing brought about the demise of the typewriter, and our department secretaries no longer prepared our research manuscripts and reports. It is interesting to us that computers are not so substantively different from other tools such that we should disregard much of what the study of human factors and experimental psychology has contributed to our understanding of human behavior in simple and complex systems. Rather, it is the computer's capacity for displaying, storing, processing, and even controlling information that has led us to the point at which the manner with which we interact with such systems has become a research area in itself.

In our studies of human-computer interaction (HCI), also known as human-machine interaction, and perceptual-motor interaction in general, we have adopted two basic theoretical and analytical frameworks as part of an integrated approach. In the first framework, we view perceptual-motor interaction in the context of an information-processing model. In the second framework, we have used analytical tools that allow detailed investigations of both static and dynamic interactions. Our chapter in the previous edition of this handbook (Chua, Weeks, & Goodman, 2003) reviewed both avenues of research and their implications for HCI with a particular emphasis on our work regarding the translation of perceptual into motor space. Much of our more recent research, however, has explored the broader interplay between the processes of action and attention. Thus, in the present chapter, we turn our focus to aspects of this work that we believe to have considerable implications for those working in HCI.

Human Information Processing and Perceptual-Motor Behavior

The information-processing framework has traditionally provided a major theoretical and empirical platform for many scientists interested in perceptual-motor behavior. The study of perceptual-motor behavior within this framework has inquired into such issues as the information capacity of the motor system (e.g., Fitts, 1954), the attentional demands of movements (e.g., Posner & Keele, 1969), motor memory (e.g., Adams & Dijkstra, 1966), and processes of motor learning (e.g., Adams, 1971). The language of information processing (e.g., Broadbent, 1958) has provided the vehicle for discussions of mental and computational operations of the cognitive and perceptual-motor system (Posner, 1982). Of interest in the study of perceptual-motor behavior is the nature of the cognitive processes that underlie perception and action.

The information-processing approach describes the human as an active processor of information, in terms that are now commonly used to describe complex computing mechanisms. An information-processing analysis describes observed behavior in terms of the encoding of perceptual information, the manner in which internal psychological subsystems utilize the encoded information, and the functional organization of these subsystems. At the heart of the human cognitive system are processes of information transmission, translation, reduction, collation, storage, and retrieval (e.g., Fitts, 1964; Marteniuk, 1976; Stelmach, 1982; Welford, 1968). Consistent with a general model of human information processing (e.g., Fitts & Posner, 1967), three basic processes have been distinguished historically. For our purposes, we refer to these processes as stimulus identification, response selection, and response programming. Briefly, stimulus identification is associated with processes responsible for the perception of information. Response selection pertains to the translation between stimuli and responses and the selection of a response. Response programming is associated with the organization of the final output (see Proctor & Vu, 2003, or the present volume).

A key feature of early models of information processing is the emphasis upon the cognitive activities that precede action (Marteniuk, 1976; Stelmach, 1982). From this perspective, action is viewed only as the end result of a complex chain of information-processing activities (Marteniuk, 1976). Thus, chronometric measures, such as reaction time and movement time, as well as other global outcome measures, are often the predominant dependent measures. However, even a cursory examination of the literature indicates that time to engage a target has been a primary measure of interest. For example, a classic assessment of perceptual-motor behavior in the context of HCI and input devices was conducted by Card, English, and Burr (1978; see also English, Engelhart, & Berman, 1967). Employing measures of error and speed, Card et al. (1978) had subjects complete a cursor positioning task using four different control devices (mouse, joystick, step keys, text keys). The data revealed the now well-known advantage for the mouse. Of interest is that the speed measure was decomposed into "homing" time, the time that it took to engage the control device and initiate cursor movement, and "positioning" time, the time to complete the cursor movement. Although the mouse was actually the poorest device in terms of the homing time measure, the advantage in positioning time produced the faster overall time. That these researchers sought to glean more information from the time measure acknowledges the importance of the movement itself in perceptual-motor interactions such as these.

The fact that various pointing devices depend on hand movement to control cursory movement has led to the emphasis that researchers in HCI have placed on Fitts' law (Fitts, 1954) as a predictive model of time to engage a target. The law predicts pointing (movement) time as a function of the distance to and width of the target—where, in order to maintain a given level of accuracy, movement time must increase as the distance of the movement increase and/or the width of the target decreases. The impact of Fitts' law is most evident by its inclusion in the battery of tests to evaluate computer-pointing devices in ISO 9241-9. We argue that there are a number of important limitations to an exclusive reliance on Fitts' law in this context.

First, although the law predicts movement time, it does this based on distance and target size. Consequently, it does not al-

low for determining what other factors may influence movement time. Specifically, Fitts' law is often based on a movement to a single target at any given time (although it was originally developed using reciprocal movements between two targets). However, in most HCI and graphical user interface (GUI) contexts, there is an array of potential targets that can be engaged by an operator. As we will discuss later in this chapter, the influence of these distracting nontarget stimuli on both the temporal and physical characteristics of the movements to the imperative target can be significant.

Second, we suggest that the emphasis on Fitts' law has diverted attention from the fact that cognitive processes involving the selection of a potential target from an array are an important, and time consuming, information processing activity that must precede movement to that target. For example, the Hick-Hyman law (Hick, 1952; Hyman, 1953) predicts the decision time required to select a target response from a set of potential responses—where the amount of time required to choose the correct response increases with the number of possible alternative responses. What is important to understand is that the two laws work independently to determine the total time it takes for an operator to acquire the desired location. In one instance, an operator may choose to complete the decision-making and movement components sequentially. Under these conditions, the total time to complete the task will be the sum of the times predicted by the Hick-Hyman and Fitts' laws. Alternatively, an operator may opt to make a general movement that is an approximate average of the possible responses and then select the final target destination while the movement is being completed. Under such conditions, Hoffman and Lim (1997) reported interference between the decision and movement component that was dependent on their respective difficulties (see also Meegan & Tipper, 1998).

Finally, although Fitts' law predicts movement time given a set of movement parameters, it does not actually reveal much about the underlying movement itself. Indeed, considerable research effort has been directed toward revealing the movement processes that give rise to Fitts' law. For example, theoretical models of limb control have been forwarded that propose that Fitts' law emerges as a result of multiple submovements (e.g., Crossman & Goodeve, 1963/1983), or as a function of both initial movement impulse variability and subsequent corrective processes late in the movement (Meyer, Abrams, Kornblum, Wright, & Smith, 1988). These models highlight the importance of conducting detailed examinations of movements themselves as a necessary complement to chronometric explorations.

For these reasons, HCI situations that involve dynamic perceptual-motor interactions may not be best indexed merely by chronometric methods (cf., Card et al., 1978). Indeed, as HCI moves beyond the simple key press interfaces that are characteristic of early systems to include virtual and augmented reality, teleoperation, gestural, and haptic interfaces, among others, the dynamic nature of perceptual-motor interactions are even more evident. Consequently, assessment of the actual movement required to engage such interfaces would be more revealing.

To supplement chronometric explorations of basic perceptual-motor interactions, motor behaviour researchers have also advocated a movement-process approach (Kelso, 1982). The argument is that, in order to understand the nature of movement organization and control, analyses should also encompass the movement itself, and not just the activities preceding it (e.g., Kelso, 1982; 1995; Marteniuk, MacKenzie, & Leavitt, 1988). Thus, investigators have examined the kinematics of movements in attempts to further understand the underlying organization involved (e.g., Brooks, 1974; Chua & Elliott, 1993; Elliott, Carson, Goodman, & Chua, 1991; Kelso, Southard, & Goodman, 1979; MacKenzie, Marteniuk, Dugas, Liske, & Eickmeier, 1987; Marteniuk, MacKenzie, Jeannerod, Athenes, & Dugas, 1987). The relevance of this approach will become apparent in later sections.

Translation, Coding, and Mapping

As outlined above, the general model of human information processing (e.g., Fitts & Posner, 1967) distinguishes three basic processes: stimulus identification, response selection, and response programming. While stimulus identification and response programming are functions of stimulus and response properties, respectively, response selection is associated with the translation between stimuli and responses (Welford 1968).

Translation is the seat of the human "interface" between perception and action. Moreover, the effectiveness of translation processes at this interface is influenced to a large extent by the relation between perceptual inputs (e.g., stimuli) and motor outputs (e.g., responses). Since the seminal work of Fitts and colleagues (Fitts & Seeger, 1953; Fitts & Deninger, 1954), it has been repeatedly demonstrated that errors and choice reaction times to stimuli in a spatial array decrease when the stimuli are mapped onto responses in a spatially compatible manner. Fitts and Seeger (1953) referred to this finding as stimulus-response (S-R) compatibility and ascribed it to cognitive codes associated with the spatial locations of elements in the stimulus and response arrays. Presumably, it is the degree of coding and recoding required to map the locations of stimulus and response elements that determine the speed and accuracy of translation and thus response selection (e.g., Wallace, 1971).

The relevance of studies of S-R compatibility to the domain of human factors engineering is paramount. It is now well understood that the design of an optimal human-machine interface in which effective S-R translation facilitates fast and accurate responses is largely determined by the manner in which stimulus and response arrays are arranged and mapped onto each other (e.g., Bayerl, Millen, & Lewis, 1988; Chapanis & Lindenbaum, 1959; Proctor & Van Zandt, 1994). As a user, we experience the recalibrating of perceptual-motor space when we take hold of the mouse and move it in a fairly random pattern when we interact with a computer for the first time. Presumably, what we are doing here is attempting to calibrate our actual movements to the resulting virtual movements of the cursor on the screen. Thus, for optimal efficiency of functioning, it seems imperative that the system is designed to require as little recalibration as possible. Again, our contribution to the previous edition of this handbook reviews our work in the area of stimulus-response translation and the implications of this work for HCI (Chua et al., 2003). We encourage those who are more interested in these issues to read that chapter.

PERCEPTUAL-MOTOR INTERACTION: ATTENTION AND PERFORMANCE

The vast literature on selective attention and its role in the filtering of target from nontarget information (e.g. Cherry, 1953; Treisman, 1964a, 1964b, 1986; Deutsch & Deutch, 1963; Treisman & Gelade, 1980) has no doubt been informative in the resolution of issues in HCI pertaining to stimulus displays and inputs (e.g., the use of color and sound). However, attention should not be thought of as a unitary function, but rather as a set of information processing activities that are important for perceptual, cognitive, and motor skills. Indeed, the evolution of HCI into the realm of augmented reality, teleoperation, gestural interfaces, and other areas that highlight the importance of dynamic perceptual-motor interactions, necessitates a greater consideration of the role of attention in the selection and execution of action. Recent developments in the study of how selective attention mediates perception and action and, in turn, how intended actions influence attentional processes, are poised to make just such a contribution to HCI. We will now turn to a review of these developments and some thoughts on their potential relevance to HCI.

Attention

We are all familiar with the concept of attention on a phenomenological basis. Even our parents, who likely never formally studied cognition, demonstrated their understanding of the essential characteristics of attention when they directed us to pay attention when we were daydreaming or otherwise not doing what was asked. They knew that humans, like computers, have a limited capacity to process information in that we can only receive, interpret, and act upon a fixed amount of information at any given moment. As such, they knew that any additional, nontask processing would disrupt the performance of our goal task, be it homework, cleaning, or listening to their lectures. But what is attention? What does it mean to pay attention? What influences the direction of our attention? The answers to these questions are fundamental to understanding how we interact with our environment. Thus, it is paramount for those who are involved in the design of HCI to consider the characteristics of attention and its interactive relationship with action planning.

Characteristics of Attention

Attention is the collection of processes that allow us to dedicate our limited information processing capacity to the purposeful (cognitive) manipulation of a subset of available information. Stated another way, attention is the process through which information enters into working memory and achieve the level of consciousness. There are three important characteristics of attention (a) attention is selective and allows only a specific subset of information to enter the limited processing system; (b) the focus of attention can be shifted from one source of information to another; and, (c) attention can be divided such that, within certain limitations, one may selectively attend to more that one source of information at a time. The well-known "cocktail party" phenomenon (Cherry, 1953) effectively demonstrates these characteristics.

Picture yourself at the last busy party or poster session you attended where there was any number of conversations continuing simultaneously. You know from your own experience that you are able to filter out other conversations and selectively attend to the single conversation in which you are primarily engaged. You also know that there are times when your attention is drawn to a secondary conversation that is continuing nearby. These shifts of attention can occur automatically, especially if you hear your name dropped in the second conversation, or voluntarily, especially when your primary conversation is boring. Finally, you know that you are able to divide your attention and follow both conversations simultaneously. However, although you are able to keep track of each discussion simultaneously, you will note that your understanding and contributions to your primary conversation diminish as you dedicate more and more of your attentional resources to the secondary conversation. The diminishing performance in your primary conversation is, of course, an indication that the desired amount of information processing has exceeded your limited capacity.

What does the "cocktail party" phenomenon tell us about designing HCI environments? The obvious implication is that, in order to facilitate the success of the performer, the HCI designer must be concerned about limiting the stress on the individuals' information processing systems by (a) creating interfaces that assist in the selection of the most appropriate information; (b) being knowledgeable about the types of attention shifts and about when (or when not) to use them; and (c) understanding that, when attention must be divided amongst a series of tasks, that each of these tasks should be designed to facilitate automatic performance so as to avoid conflicts in the division of our limited capacity and preserve task performance. While these suggestions seem like statements of the obvious, the remainder of the chapter will delve deeper into these general characteristics and highlight situations in which some aspects of design might not be as intuitive as it seems. Because vision is the dominant modality of information transfer in HCI, we will concentrate our discussion on visual selective attention. It should be noted, however, that there is a growing literature on cross-modal influences on attention, especially visual-auditory system interactions (e.g., Spence, Lloyd, McGlone, Nichols, & Driver, 2000), that will be relevant in the near future.

Shifts and Coordinate Systems of Attention

Structural analyses of the retinal (photo sensitive) surface of the eye has revealed two distinct receiving areas—the fovea and the perifoveal (peripheral) areas. The fovea is a relatively small area (about two to three degrees of visual angle) near the center of the retina, which has the highest concentration of color-sensitive cone cells. It is this high concentration of color-sensitive cells that provides the rich, detailed information that we typically use to identify objects. There are several important consequences of this structural and functional arrangement. First, because of the foveas' pivotal role in object identification and the importance of object identification for the planning of

action and many other cognitive processes, visual attention is typically dedicated to the information received by the fovea. Second, because the fovea is such a small portion of the eye, we are unable to derive a detailed representation of the environment from a single fixation. As a result, it is necessary to constantly move information from objects in the environment onto the fovea by rotating the eye rapidly and accurately. These rapid eye movements are known as saccadic eye movements. Because of the tight link between the location of visual attention and saccadic eye movements, these rapid eye movements are referred to as overt shifts of attention.

Although visual attention is typically dedicated to foveal information, it must be remembered that the perifoveal retinal surface also contains color-sensitive cells and, as such, is able to provide details about objects. A covert shift of attention refers to any situation in which attention is being dedicated to a nonfoveated area of space. Covert shifts of attention are employed when the individual wants or needs to maintain the fovea on a particular object while continuing to scan the remaining environment for other stimuli. Covert shifts of attention also occur immediately prior to the onset of an overt shift of attention or other type of action (e.g., Shepherd, Findlay, & Hockey, 1986). For this reason, people are often able to identify stimuli at the location of covert attention prior to the acquisition of that location by foveal vision (e.g., overt attention) (Deubel & Schneider, 1996).

Because attention is typically dedicated to the small foveal subdivision of the retinal surface, attention is often considered to work as a spotlight or zoom lens that constantly scans the environment (e.g., Eriksen & Eriksen, 1974). More often than not, however, the objects that we attend to are larger than the two to three degrees of visual angle covered by the fovea. Does this mean that the components of objects that are outside of foveal vision do not receive attentional processing? No, in fact it has been repeatedly shown that attention can work in an object-based coordinate system where attention actually spreads along the full surface of an object when attention is dedicated to a small section of the object (Davis & Driver, 1997; Egly, Driver, & Rafal, 1994; see also Duncan, 1984). These object-centered attentional biases occur even when other objects block connecting sections of the continuous object (e.g., Pratt & Sekuler, 2001). Finally, it should be noted that, although entire objects receive attentional processing, particular sections of the object often receive preferential attentional treatment often based on the action potential of the object (see Attention and Stimulus-Response Compatibility below). Thus, attention should be seen as a flexible resource allocation instead of a fixed commodity with a rigid structure. The spotlight coding system is typically employed during detailed discrimination tasks, for example when reading the text on this page, whereas object-based coding might be more effective when gaining an appreciation for the context of an object in the scene or the most interactive surface of the object. The relevance of the flexible, action-dependent nature of attentional coding systems will be readdressed in latter sections.

Stimulus Characteristics and Shifts of Attention

Both overt and covert shifts of attention can be driven by stimuli in the environment or by the will of the performer. Shifts of attention that are driven by stimuli are known as exogenous, or bottom-up, shifts of attention. They are considered to be automatic in nature and thus, for the most part, are outside of cognitive influences. Exogenous shifts of attention are typically caused by a dynamic change in the environment such as the sudden, abrupt appearance (onset) or disappearance (offset) of a stimulus (e.g., Pratt & McAuliffe, 2001), a change in the luminance or color of a stimulus (e.g., Folk, Remington, & Johnston, 1992; Posner, Nissen, & Ogden, 1978; Posner & Cohen, 1984), or the abrupt onset of object motion (e.g., Abrams & Chirst, 2003; Folk, Remington, & Wright, 1994). The effects of exogenous shifts have a relatively rapid onset, but are fairly specific to the location of the dynamic change and are transient, typically reaching their peak influence around 100 milliseconds after the onset of the stimulus (Cheal & Lyon, 1991; Müller & Rabbitt, 1989). From an evolutionary perspective, it could be suggested that these automatic shifts of attention developed because such dynamic changes would provide important survival information such as the sudden, unexpected appearance of a predator or prey. However, in modern times, these types of stimuli can be used to quickly draw one's attention to the location of important information.

In contrast, performer-driven, or endogenous, shifts of attention are under complete voluntary control. The effects of endogenous shifts of attention take longer to develop, but can be sustained over a much longer period of time (Cheal & Lyon, 1991; Müller & Rabbitt, 1989). From an HCI perspective, there advantages and disadvantages to the fact that shifts of attention can be under cognitive control. The main benefit of cognitive control is that shifts of attention can result from a wider variety of stimuli such as symbolic cues like arrows, numbers, or words. In this way, performers can be cued to locations or objects in the scene with more subtle or permanent information than the dynamic changes that are required for exogenous shifts. The main problem with endogenous shifts of attention is that the act of interpreting the cue requires a portion of the limited information processing capacity and thus can interfere with, or be interfered by, concurrent cognitive activity (Jonides, 1981).

Although it was originally believed that top-down processes could not influence exogenous shifts of attention (e.g., that dynamic changes reflexively capture attention regardless of intention), Folk et al. (1992) demonstrated that this is not always the case. The task in the Folk et al. (1992) study was to identify a stimulus that was presented in one of four possible locations. For some participants, the target stimulus was a single abrupt onset stimulus (the target appeared in one location and nothing appeared in the other three locations), whereas for the remaining participants the target stimulus was a color singleton (a red stimulus that was presented at the same time as white stimuli that appeared in the other three possible locations). One-hundred fifty milliseconds prior to the onset of the target, participants received cue information at one of the possible target locations. The cue information was either abrupt onset stimuli at a single location or color singleton information. Across a series of experiments, Folk et al. (1992) found that the cue tended to slow reaction times to the target stimulus when the cue information was presented at a location that was different from where the target subsequently appeared indicating that attention had initially been exogenously drawn to the cue. Impor-

tantly, the cue stimuli only interfered with the identification of the target stimulus when the characteristics of cue stimuli matched the characteristics of the target stimulus (e.g., onset cue-onset target and color cue-color target conditions). When the characteristics of the cue did not match the target stimulus (e.g., onset cue-color target and color cue-onset target conditions), the location of the cue did not influence reaction times. Thus, these results reveal that dynamic changes only capture attention when the performer is searching for a dynamic change stimulus. Stated another way, it seems that automatic attentional capture is dependent upon the expectations of the performer. Folk et al. (1992) suggested that people create an attention set in which they establish their expectations for the characteristics of the target stimulus. Stimuli meeting the established set will automatically capture attention, whereas stimuli that do not meet the established set will not (see also Folk et al., 1994). The obvious implication of these results is that the most efficient HCIs will be those for which the designer has considered perceptual expectations of the person controlling the system. As we will discuss in the Action-Centered Attention section, however, consideration of the perceptual components alone is, at best, incomplete.

Facilitation and inhibition of return. While it is the case that our attention can be endogenously and exogenously shifted to any location or object in our environment, it seems there are unconscious mechanisms that work to hinder the movement of our attention to previously investigated locations and objects. The existence of these reflexive mechanisms was first revealed through a series of studies by Posner and Cohen (1984) and has been reliably demonstrated many times since. The basic task in these studies is to respond as quickly as possible following the onset of a target that randomly appears at one of any number of possible locations. The presentation of the target is preceded by a briefly presented cue that is not predictive of the target location (e.g., if there are 2 possible target locations, the target will appear at the location of the cue on 50% of the trials and at the uncued location of the remaining 50% of the trials). The key findings of these studies are that (a) when there is a short time interval between the onset of the cue and the onset of the target (less than 200 ms), participants are faster at responding to targets presented at the cued location than at the uncued location; whereas, (b) when there is a longer time interval between the onset of the cue and the onset of the target (greater than 400–500 ms), participants are faster at responding to the target presented at the uncued location than at the cued location (see Fig. 1.1 for some representative data). It is important to remember that the reaction times to cued targets are facilitated at short intervals and inhibited at longer intervals even though the cue has no predictive relation to the location of the subsequent target. The earlier facilitation effect is thought to arise because attention has been exogenously drawn from a central fixation point to the location of the cue and is still there when the target subsequently appears—with attention already at the target location, subsequent target processing is efficient. As the time elapses after the onset of the cue, however, the performer knows that the target is equally likely to appear at any location and so endogenously returns attention back to the central point. It is suggested that in moving attention back to a central point, the performer places an inhibitory code on the location of the cue. This inhibitory code subsequently interferes with the processing of target when it appears at the location of the cue and increases reactions for the cued target relative to any uncued (e.g., uninhibited) target leading to the inhibition of return (IOR) effect.

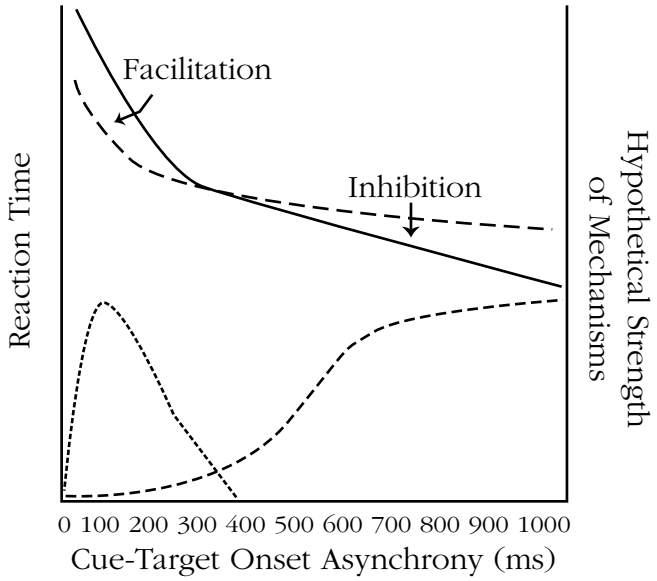


FIGURE 1.1. Exemplar reaction time data as a function of Target Location (cued [large dashed black line] vs. uncued [solid black line]) and Cue-Target Onset Asynchrony (CTOA) and hypothetical activation levels of the facilitatory [dotted line] and inhibitory [small dashed line] mechanisms that cause Facilitation and Inhibition of Return.

The twenty plus years of research on these phenomena has revealed many characteristics of the mechanisms of attention system that are relevant for HCI. First is the time course of the development of the mechanisms underlying these facilitation and IOR effects. It is thought that the mechanism of facilitation has a very short onset and a small life, whereas the mechanism of inhibition has a longer latency but has a much longer lasting influence (up to three to four seconds; see Fig. 1.1). Thus, if a designer intends on using a dynamic environment in which irrelevant, but perhaps aesthetic, stimuli constantly appear, disappear, or move, then it is important to realize that the users' performance might be negatively influenced because of the inadvertent facilitation of the identification of nontarget information or the inhibition of the identification of target information depending on the spatiotemporal relationship between the relevant and irrelevant information. Alternatively, video game designers could exploit these effects to suit their purpose when they want to facilitate or hinder the performance of a gamer in a certain situation as it has been shown that even experienced video game players demonstrate facilitation and IOR effects of similar magnitude to novice gamers (Castel, Drummond, & Pratt, 2005).

The second important feature is that facilitation and IOR only occur when attention has been shifted to the location of the cue. Thus, if the designer is using cues that typically only produce endogenous shifts of attention (e.g., by using a symbolic cue such as an arrow or word that indicates a particular location), and then reaction times will be similar across cued and uncued targets (Posner & Cohen, 1984). The lack of facilitation and IOR following symbolic cues is thought to occur because the participant can easily ignore the cue and prevent the shift of attention to the cued location that activates the facilitatory or inhibitory mechanisms. It is important to note, however, the mechanisms of IOR are activated each time attention has been shifted. Thus, symbolic information can result in IOR if the participant does shift their attention to the cued location (even if they were asked not to) or if the participant is required to respond to the cue information (Rafal, Calabresi, Brennan, & Sciolto, 1989; Taylor & Klein, 2000). Although symbolic information presented at locations that are outside of the target locations do not typically activate the possibly detrimental mechanisms of facilitation and IOR, one must still use caution when presenting such information and be sensitive to context (e.g., how similar symbolic cues have been used in previous interactions) and the response expectations of the user.

Finally, as briefly mentioned in the previous paragraph, it is important to realize that the inhibitory mechanisms of IOR are also activated by responding to a location. That is, when people are required to make a series of responses to targets that are randomly presented at one of a number of possible locations, they are slower at initiating response *n* when it is the same as response *n-1* than when it is different from response *n-1* (Maylor & Hockey, 1985; Tremblay, Welsh, & Elliott, 2005). This target-target IOR effect has important implications for two reasons. First, if the designer intends on requiring the user to complete a series of choice responses to targets that appear at the same locations each time and the user is uncertain as to where each target will appear (as in typical interactions with automated banking machines), then the user will be slower at responding to targets that appear successively at the same location than to targets that appear at new locations. When it is also considered that it has been shown that, in a cue-target IOR experiments, inhibitory codes can be maintained at up to four locations simultaneously (Tipper, Weaver, & Watson, 1996; Wright & Richard, 1996; cf., Abrams & Pratt, 1996), the designer must be cautious about designing displays that use similar locations on successive interactions. The second reason that an understanding of target-target IOR is important for HCI is that we have recently shown that target-target IOR effects transfer across people (Welsh, Elliott, Anson, Dhillon, Weeks, Lyons, et al., 2005; Welsh, Lyons, Weeks, Anson, Chua, Mendoza, et al., in press). That is, if two people are performing a task in which they must respond to a series of successive targets, then an individual will be slower at returning to the previously responded-to location regardless if that person or their partner completed the initial response. Although the research on social attention is in its infancy, the results of our work indicate that those involved in the emerging fields of virtual reality or immersive collaborative and multiuser environments must be aware of and consider how attention is influenced by a variety of stimuli including the actions of other participants.

Action-Centered Attention

The majority of the literature reviewed thus far has involved experiments that investigated attentional processes through tasks that employed simple or choice key press actions. Cognitive scientists typically use these arbitrary responses (a) because key press responses are relatively uncomplicated and provide simple measures of performance, namely reaction time and error; and (b) because, by using a simple response, the researcher assumes that they have isolated the perceptual and attentional processes of interest from additional, complex motor programming and control processes. While there are certainly numerous examples of HCIs in which the desired response is an individual key press or series of key presses, there are perhaps as many situations in which movements that are more complicated are required. As HCIs move increasingly into virtual reality, touch screen, and other more complex environments, it will become more and more important to consider the ways in which attention and motor processes interact. Thus, it will become more and more critical to determine if the same principles of attention apply when more involved motor responses are required. In addition, some cognitive scientists have suggested that, because human attention systems have developed through evolution to acquire the information required to plan and control complex actions, studying attention under such constrained response conditions may actually provide an incomplete or biased view of attention (Allport, 1987; 1993). The tight link between attention and action is apparent when one recognizes that covert shifts of attention occur prior to saccadic eye movements (Deubel & Schneider, 1996) and that overt shifts of attention are tightly coupled to manual aiming movements (Helsen, Elliott, Starkes, & Ricker, 1998; 2000). Such considerations, in combination with neuroanatomical studies revealing tight links between the attention and motor centers (Rizzolatti, Riggio, & Sheliga, 1994), have lead to the development of action-centered models of attention (Rizzolatti, Riggio, Dascola, & Umilta, 1987; Tipper, Howard, & Houghton, 1999; Welsh & Elliott, 2004a).

Arguably the most influential work in the development of the action-centered models was the paper by Tipper, Lortie, and Baylis (1992). Participants in these studies were presented with nine possible target locations, arranged in a three-by-three matrix and were asked to identify the location of a target stimulus appearing at one of these locations while ignoring any nontarget stimuli presented at one of the remaining eight locations. The key innovation of this work was that Tipper and colleagues (1992) asked participants to complete a rapid aiming movement to the target location instead of identifying it with a key press. Consistent with traditional key press studies, the presence of a distractor was found to increase response times to the target. Although the finding of distractor interference in this selective reaching task was an important contribution to the field in and of itself, the key discovery was that the magnitude of the interference effects caused by a particular distractor location was dependent on the aiming movement being completed. Specifically, it was found that distractors (a) along the path of the movement caused more interference than those that were outside the path of the movement (the proximity-to-hand effect);

and, (b) ipsilateral to the moving hand caused more interference than those in the contralateral side of space (the ipsilateral effect). Based on this pattern of interference, Tipper et al. (1992) concluded that attention and action are tightly linked such that the distribution of attention is dependent on the action that was being performed (e.g., attention was distributed in an action-centered coordinate system) and that the dedication of attention to a stimulus evokes a response to that stimulus.

While the Tipper et al. (1992) paper was critical in initiating investigations into action-centered attention, more recent research has demonstrated that the behavioral consequences of selecting and executing target-directed actions in the presence of action-relevant nontarget stimuli extend beyond the time taken to prepare and execute the movement (e.g., Meegan & Tipper, 1998; Pratt & Abrams, 1994). Investigations in our labs and others have revealed that the actual execution of the movement changes in the presence of distractors. For example, there are reports that movements will deviate towards (Welsh, Elliott, & Weeks, 1999; Welsh & Elliott, 2004a; 2005) or away from (Howard & Tipper, 1997; Tipper, Howard, & Jackson, 1997; Welsh & Elliott, 2004a) the nontarget stimulus. Although these effects seem paradoxical, Welsh and Elliott (2004a) have formulated and tested (e.g., Welsh & Elliott, 2004a; 2004b; 2005) a conceptual model that can account for these movement execution effects.

The model of response activation. Consistent with the conclusions of Tipper et al. (1992), Welsh and Elliott (2004a) based the model of response activation on the premise that attention and action processes are so tightly linked that the dedication of attention to a particular stimulus automatically initiates response-producing processes that are designed to interact with that stimulus. Responses are activated to attended stimuli regardless of the nature of attentional dedication (e.g., reflexive or voluntary). It is proposed that each time a performer approaches a known scene, a response set is established in working memory in which the performer identifies and maintains the characteristics of the expected target stimulus and the characteristics of the expected response to that stimulus. Thus, the response set in the model of response activation is an extension of the attentional set of Folk et al. (1992) in that the response set includes the performer's expectations of the target stimulus as well as preexcited (preprogrammed) and/or preinhibited response codes. Each stimulus that matches the physical characteristics established in the response set captures attention and, as a result, activates an independent response process. Stimuli that do not possess at least some of the expected characteristics do not capture attention and thus do not activate responses. Thus, if only one stimulus in the environment matches the response set, then that response process is completed unopposed and the movement emerges rapidly and in an uncontaminated form. On the other hand, under conditions in which more than one stimulus matches the response set, multiple response representations are triggered and subsequently race one another to surpass the threshold level of neural activation required to initiate a response. It is important to note that this is not a winner-take-all race where only the characteristics of the winning response influence the characteristics of actual movement alone. Instead, the characteristics of the observed movement are determined by the activation level of each of the competing responses at the moment of movement initiation. In this way, if more than one neural representation is active (or if one is active and one is inhibited) at response initiation, then the emerging response will have characteristics of both responses (or characteristics that are opposite to the inhibited response).

The final relevant element of the model is that the activation level of each response is determined by at least three interactive factors: the salience of the stimulus and associated response, an independent inhibitory process, and the time course of each independent process. The first factor, the salience or action-relevancy of the stimulus, is in fact the summation of a number of separate components including the degree attentional capture (based on the similarity between the actual and anticipated stimulus within the response set), the complexity of the response afforded by the stimulus, and the S-R compatibility. When attentional capture and stimulus-response compatibility are maximized and response complexity minimized, the salience of an individual response is maximized and the response to that stimulus will be activated rapidly. The second factor that influences the activation level of a response is an independent inhibitory process that works to eliminate nontarget responses. When this inhibitory mechanism has completed its task, the neuronal activity associated with the inhibited response will have been reduced to below baseline. Thus, in effect, an inhibited response will add characteristics that are opposite to it to the formation of the target response. The final factor that contributes to the activation level of each independent response process is the time course of the development of the representation. It is assumed that the neural representations of each response do not develop instantaneously and that the inhibitory mechanism that eliminates nontarget responses does not instantly remove the undesired neural activity. Instead, each of these processes requires time to reach an effective level and the time course of each responses' development will be independent of one another. For example, a response to a very salient stimulus will achieve a higher level of activation sooner than a response to a stimulus with a lower saliency. If this stimulus of high salience evokes the desired response, then responses to other, less salient stimuli will cause little interference because they simply do not reach as high an activation level than the target response. In contrast, when nontarget stimuli are more salient, then the interference is much more severe (Welsh & Elliott, 2005). In sum, to extend the analogy of the race model, responses that run faster (the responses were evoked by more salient stimuli), have a head start relative to another (one stimulus was presented prior to another), or have a shorter distance to go to the finish line (the response was partially preprogrammed in the response set) will achieve a higher level of activation and will, as a result, contribute more to the characteristics of the final observable movement than ones that are further behind.

So, what implications does the model of response activation have for the design of HCI? In short, because the model of response activation provides a fairly comprehensive account of movement organization in complex environments, it could be used as the basis for the design of interfaces that consider the cognitive system as an interactive whole as opposed to separate units of attention and movement organization. One of the

more obvious implications is that a designer should consider the time intervals between the presentation of each stimulus in a multiple stimuli set as this can have dramatic effects on the performers ability to quickly respond to each stimulus (e.g., psychological refractory period; Telford, 1931; Pashler, 1994) and the physical characteristics of each response (Welsh & Elliott, 2004a).

Perhaps more importantly, the model highlights the importance for the designer to consider the interactions between attention and motor processing because there are some situations in which the transfer from simple to complex movements is straightforward, whereas there are others that do not transfer at all. The study by Bekkering and Pratt (2004) provided a good demonstration of a situation in which the interaction between attention and action provides the same results in key press and aiming responses. Specifically, they showed that a dynamic change on one location of an object facilitates reaction times for aiming movements to any other portion of the object (e.g., attention can move in object-centered coordinate systems in aiming responses as it does for key press response). However, Welsh & Pratt (2005) recently found that the degree of attentional capture by some dynamic changes is different when key press and spatially-directed responses are required. In this study, participants were asked to identify the location of an onset or offset target stimulus while ignoring a distractor stimulus of the opposite characteristics (e.g., onset targets were paired with offset distractors and vice versa). In separate experiments, participants responded to the target stimulus with a choice key press response or an aiming movement to the target location. The results indicated that an onset distractor slowed responding to an offset target in both tasks. Offset distractor, on the other hand, only interfered with task performance when a key press was required. When participants were asked to perform an aiming movement, an offset distractor actually caused a nonsignificant facilitation effect. Similar action-specific interference effects have been shown across pointing and grasping actions (Bekkering & Neggers, 2002), pointing and verbal responses (Meegan & Tipper, 1999), and different types of pointing responses (Meegan & Tipper, 1999; Tipper, Meegan, & Howard, 2002). In sum, now that HCI is moving into virtual reality and other types of assisted response devices (voice activated, head mounted, roller ball, and eye-gaze mouse systems), it will become increasingly important to consider the required and/or anticipated action when designing HCI environments. Given our emphasis on the importance of considering the interaction of attention and motor processes, we will explore this issue in greater detail in the following sections.

Attention and action requirements. Initial investigations into action-centered attention were focused primarily on the influence that the spatial location of distractors with respect to the target had on the planning and execution of action (e.g., Meegan & Tipper, 1998; Pratt & Abrams, 1994; Tipper et al., 1992). In that context, an action-centered framework could offer a useful perspective for the spatial organization of perceptual information presented in an HCI context. However, often the reason for engaging a target in an HCI task is that the target symbolically represents an outcome or operation to be achieved. Indeed, this is what defines an icon as a target—target features symbolically carry a meaning that defines it as the appropriate target. An interest in the application of the action-centered model to human factors and HCI led Weir, Weeks, Welsh, Elliott, Chua, Roy, and Lyons (2003) to consider whether or not distractor effects could be elicited based upon the specific actions required to engage a target and distractor object. The question was whether the engagement properties of target and distractor objects (e.g., turn or pull) in a control array would mediate the influence of the distractor on the control of movement. In that study, participants executed their movements on a control panel that was located directly in front of them. On some trials, the control panel consisted of a single pull-knob or right-turn dial located at the midline either near or far from a starting position located proximal to the participant. On other trials, a second control device (pull knob or dial) was placed into the other position on the display. If this second device was present, it served as a distractor object and was to be ignored. Weir et al. (2003) found that the distractor object only interfered with the programming of the response to the target stimulus when the distractor afforded a different response from the target response (e.g., when the pull knob was presented with the turn dial). These results suggest that when moving in an environment with distracting stimuli or objects, competing responses may be programmed in parallel and that these parallel processes will only interfere with one another when they are incompatible. The implication is that the terminal action required to engage the objects in the environment is also important to the distribution of attention and movement planning and execution.

In addition to considering the terminal action requirements, other work in our labs suggests that the manner in which the actions are completed is an equally important concern. For example, the "cluttered" environment of response buttons employed by researchers interested in selective reaching struck us as being analogous to the array of icons present in a typical GUI. In a study, Lyons, Elliott, Ricker, Weeks, and Chua (1999) sought to determine whether these paradigms could be imported into a "virtual" environment and ultimately serve as a test bed for investigations of perceptual-motor interactions in an HCI context. The task space in the Lyons et al. (1999) study utilized a three-by-three matrix similar to that used by Tipper et al. (1992). The matrix, made up of nine blue circles, was displayed on a monitor placed vertically in front of the participant. Participants were required to move the mouse on the graphics tablet, which would in turn move a cursor on the monitor in the desired direction toward the target (red) circle while ignoring any distractor (yellow) circles. The participants were unable to view their hand; the only visual feedback of their progress was from the cursor moving on the monitor. The graphics tablet allowed the researchers to record displacement and time data of the mouse throughout the trial. In contrast to previous experiments (e.g., Meegan & Tipper, 1998; Tipper et al., 1992), the presence of a distractor had relatively little influence on performance. Lyons et al. (1999) postulated that, in a task environment in which perceptual-motor interaction is less direct (e.g., using a mouse to move a cursor on a remote display) perceptual and motor workspaces are misaligned, and the increased translation processing owing to the misalignment serves to limit the impact of distractor items.

To test this idea, Lyons et al. (1999) modified the task environment so as to align the perceptual and motor workspaces. The monitor was turned and held screen down inside a support frame. The same three-by-three matrix was displayed on the monitor and reflected into a half-silvered mirror positioned above the graphics tablet allowing for sufficient space for the participant to manipulate the mouse and move the cursor to the target without vision of the hand. With this configuration, the stimulus display was presented and superimposed on the same plane as the motor workspace (e.g., the graphics tablet). Under this setup, distractor effects emerged and were consistent with an action-centered framework of attention. Taken together, these findings underscore the influence of translation requirements demanded by relative alignment of perceptual and motor workspaces. More importantly, these findings suggest that even relatively innocuous changes to the layout of the task environment may have significant impact on processes associated with selective attention in the mediation of action in an HCI context.

Attention and stimulus-response compatibility. Thus far we have only lightly touched on the issue of attention and S-R compatibility. However, the action-centered model of selective attention clearly advocates the view that attention and action are intimately linked. The fundamental premise is that attention mediates perceptual-motor interactions, and these, in turn, influence attention. Consistent with this perspective, the role of attention in the translation between perceptual inputs and motor outputs has also received considerable interest over the past decade. As discussed above, a key element in the selection of an action is the translation between stimuli and responses, the effectiveness of which is influenced to a large extent by the spatial relation between the stimuli and responses. The degree of coding and recoding required to map the locations of stimulus and response elements has been proposed to be a primary determinant of the speed and accuracy of translation (e.g. Wallace, 1971). Attentional processes have been implicated in the issue of how relative spatial stimulus information is coded. Specifically, the orienting of attention to the location of a stimulus has been proposed to result in the generation of the spatial stimulus code.

Initial interest in the link between attention orienting and spatial translation have emerged as a result of attempts to explain the Simon effect. The Simon effect (Simon, 1968; Simon & Rudell, 1969), often considered a variant of spatial S-R compatibility, occurs in a situation in which a nonspatial stimulus attribute indicates the correct response and the spatial attribute is irrelevant to the task. Thus, the spatial dimension of the stimulus is an irrelevant attribute, and a symbolic stimulus feature constitutes the relevant attribute. Although the spatial stimulus attribute is irrelevant to the task, faster responding is found when the position of the stimulus and the position of the response happen to correspond. A number of researchers (e.g., Umiltà & Nicoletti, 1992) have suggested that attentional processes may be a unifying link between the Simon effect and the spatial compatibility effect proper. Specifically, the link between attention and action in these cases is that a shift in attention is postulated to be the mechanism that underlies the generation of the spatial stimulus code (e.g., Nicoletti & Umiltà, 1989, 1994; Proctor & Lu, 1994; Rubichi, Nicoletti, Iani, & Umiltà, 1997; Stoffer, 1991; Stoffer & Umiltà, 1997; Umiltà & Nicoletti, 1992). According to an attention-shift account, when a stimulus is presented to the left or right of the current focus of attention, a reorienting of attention occurs toward the location of the stimulus. This attention shift is associated with the generation of a spatial code that specifies the position of the stimulus with respect to the last attended location. If this spatial stimulus code is congruent with the spatial code of the response, then S-R translation, and therefore response selection, is facilitated. If the two codes are incongruent, response selection is hindered.

Recent work in our lab has also implicated a role for attention shifts in compatibility effects and object recognition. In these studies, Lyons, Weeks, and Chua (2000a; 2000b) sought to examine the influence of spatial orientation on the speed of object identification. Participants were presented with video images of common objects that possessed a graspable surface (e.g., a tea cup, frying pan), and were instructed to make a left or right key press under two distinct mapping rules depending on whether the object was in an upright or inverted vertical orientation. The first mapping rule required participants to respond with a left key press when the object was inverted and a right key press when the object was upright. The opposite was true for the second mapping rule. The orientation of the object's graspable surface was irrelevant to the task. The results showed that identification of object orientation was facilitated when the graspable surface of the object was also oriented to the same side of space as the response (see also Tucker & Ellis, 1998). In contrast, when participants were presented with objects that possessed symmetrical graspable surfaces on both sides (e.g., a sugar bowl with two handles), identification of object orientation was not facilitated. Lyons et al. (2000b) also showed that response facilitation was evident when the stimuli consisted simply of objects that, though may not inherently be graspable, possessed a left-right asymmetry. Taken together, these results were interpreted in terms of an attentional mechanism. Specifically, Lyons et al. (2000a; 2000b) proposed that a left-right object asymmetry (e.g., a protruding handle) might serve to capture spatial attention (cf., Tucker & Ellis, 1998). If attention is thus oriented toward the same side of space as the ensuing action, the spatial code associated with the attention shift (e.g., see discussion above) would lead to facilitation of the response. In situations in which no such object asymmetry exists, attentional capture and orienting may be hindered, and as a result, there is no facilitation of the response.

Taken into the realm of HCI, it is our position that the interplay between shifts of attention, spatial compatibility, and object recognition will be a central human performance factor as technological developments continue to enhance the directness of direct-manipulation systems (cf., Shneiderman, 1983; 1992). Specifically, as interactive environments become better abstractions of reality with greater transparency (Rutkowski, 1982), the potential influence of these features of human information-processing will likely increase. Thus, it is somewhat ironic that the view toward virtual reality, as the solution to the problem

of creating the optimal display representation, may bring with it an "unintended consequence" (Tenner, 1996). Indeed, the operator in such an HCI environment will be subject to the same constraints that are present in everyday life.

The primary goal of human factors research is to guide technological design in order to optimize perceptual-motor interactions between human operators and the systems they use within the constraints of maximizing efficiency and minimizing errors. Thus, the design of machines, tools, interfaces, and other sorts of devices utilizes knowledge about the characteristics, capabilities, as well as limitations, of the human perceptual-motor system. In computing, the development of input devices such as the mouse and graphical user interfaces was intended to improve human-computer interaction. As technology has continued to advance, the relatively simple mouse and graphical displays have begun to give way to exploration of complex gestural interfaces and virtual environments. This development may be, perhaps in part, a desire to move beyond the artificial nature of such devices as the mouse, to ones that provide a better mimic of reality. Why move an arrow on a monitor using a hand-held device to point to a displayed object, when instead, you can reach and interact with the object? Perhaps such an interface would provide a closer reflection of real-world interactions—and the seeming ease with which we interact with our environments, but also subject to the constraints of the human system.

PERCEPTUAL-MOTOR INTERACTION IN APPLIED TASKS: A FEW EXAMPLES

As we mentioned at the outset of this chapter, the evolution of computers and computer-related technology has brought us to the point at which the manner with which we interact with such systems has become a research area in itself. Current research in motor behavior and experimental psychology pertaining to attention, perception, action, and spatial cognition is poised to make significant contributions to the area of HCI. In addition to the continued development of a knowledge base of fundamental information pertaining to the perceptual-motor capabilities of the human user, these contributions will include new theoretical and analytical frameworks that can guide the study of HCI in various settings. In this final section, we highlight just a few specific examples of HCI situations that offer a potential arena for the application of the basic research that we have outlined in this chapter.

Remote Operation and Endoscopic Surgery

Work by Hanna, Shimi, and Cuschieri (1998) examined task performance of surgeons as a function of the location of the image display used during endoscopic surgical procedures. In their study, the display was located in front, to the left, or to the right of the surgeon. In addition, the display was placed either at eye level or at the level of the surgeon's hands. The surgeons' task performance was observed with the image display positioned at each of these locations. Hanna et al. (1998) showed that the surgeons' performance was affected by the location of the display. Performance was facilitated when the surgeons were allowed to view their actions with the monitor positioned in front and at the level of the immediate workspace (the hands). Similar findings have also been demonstrated by Mandryk and MacKenzie (1999). In addition to the frontal image display location employed by Hanna et al. (1998), Mandryk and MacKenzie (1999) also investigated the benefits of projecting and superimposing the image from the endoscopic camera directly over the workspace. Their results showed that performance was superior when participants were initially exposed to the superimposed viewing conditions. This finding was attributed to the superimposed view allowing the participants to better calibrate the display space with the workspace. These findings are consistent with our own investigations of action-centered attention in virtual environments (Lyons et al., 1999). We would suggest that the alignment of perceptual and motor workspaces in the superimposed viewing condition facilitated performance due to the decreased translation requirements demanded by such a situation. However, the findings of Lyons et al. (1999) would also lead us to suspect that this alignment may have additional implications with respect to processes associated with selective attention in the mediation of action. Although the demands on perceptual-motor translation may be reduced, the potential intrusion of processes related to selective attention and action selection may now surface. Thus, we would cautiously suggest that the optimal display location in this task environment placed less translation demands on the surgeon during task performance.

In addition to considering the orientation of the workspace of the surgeon, it must be recognized that the surgeons work with a team of support personnel performing a variety of actions all designed to achieve a common goal. Thus, the group of people can be considered to be a synergistic unit much like an individual consists of a synergistic group of independent processes. This similarity in functional structure led us (Welsh et al., 2005, in press) and others to contemplate whether the behavior of a group of people in perceptual-motor and attention tasks follow the same principles as those that govern the behavior of an individual. While these initial experiments provide converging evidence that the same processes are involved in group and individual behavior, this line of research is still in its infancy and additional research is required to determine possible conditions in which different rules apply. For example, an interesting qualification is that the interactions between attention and action that occur when two humans interact do not emerge when a human interacts with a robotic arm (Castiello, 2003). Thus, given the different vantage points and goals of each member of the surgical team, an important empirical and practical question is will be the manner in which perceptual-motor workspace can be effectively optimized to maximize the success of the team.

Personal Digital Assistants

Hand-held computer devices (PDAs and other similar communication devices like the Blackberry) are becoming increasingly

sophisticated as they become more and more popular as mobile information processing and communication systems. One of the very relevant aspects of the evolving design of PDAs is the incorporation of a stylus touch screen system. This design allows for a tremendous flexibility in possible functions while at the same time maintains a consistent coding between real and virtual space. This latter advantage should allow for the straightforward application of the principles of action and attention discussed in this chapter to this virtual environment (e.g., Tipper et al., 1992 vs. Lyons et al., 1999). However, caution should once more be urged when implementing design change without due consideration of issues such as S-R compatibility as it has been shown that novices are fairly poor judges of configurations that facilitate the most efficient performance (Payne, 1995; Vu & Proctor, 2003). In addition to the principles of action-centered attention for lower-level interactions such as program navigation, the design of higher-order language-based functions must also take into account the experience and expectations of the user. For example, Patterson and Lee (2005) found that participants had greater difficulty in learning to use characters from the new written language developed for some PDAs when those characters did not resemble the characters of traditional English. Thus, as been highlighted many times in this chapter, consideration of perceptual and motor expectations of the user is necessary for the efficient use of these devices.

Eye-Gaze vs. Manual Mouse Interactions

Based on our review of attention and action, it is obvious that the spatiotemporal arrangement of relevant and nonrelevant stimulus information will affect the manner in which actions are completed. Although the work conducted in our labs (e.g., Lyons et al., 1999; Welsh et al., 1999; Welsh & Elliott, 2004a; 2004b) revealed that manual responses, such as those controlling conventional mouse or touch screen systems, are affected by nontarget distracting information, research from other labs has revealed that eye movements are similarly affected by the presentation of nontarget information (e.g., Godjin & Theewues, 2004). Given the intimate link between eye movements and attention, this similarity between the action-centered effects on eye and hand movements should not be surprising. Although the similarities in the interactions between attention and the manual and eye movement systems would initially lead one to believe that environments designed for manual responses can be immediately imported into eye-gaze controlled systems, there are important differences between the systems that must be considered. For example, it has been reported that one of the fundamental determinants of manual reaction times, the number of possible target locations (Hick-Hyman law; Hick, 1952; Hyman, 1953) does not influence eye movement reaction times (Kveraga, Boucher, & Hughes, 2002). Thus, while the programming of eye-gaze controlled HCI systems may be much more complicated than those involved in manual systems, the productivity of the user of eye-gaze systems may be more efficient than those of traditional manual (mouse) systems (Sibert & Jacob, 2000). Although eye movements still seem to be susceptible to the goals of the manual system (Bekkering & Neggers, 2002), additional investigation into the viability of eye-gaze and/or interactive eye-gaze and manual response systems is certainly warranted.

SUMMARY

The field of HCI offers a rich environment for the study of perceptual-motor interactions. The design of effective human-computer interfaces has been, and continues to be, a significant challenge that demands an appreciation of the entire human perceptual-motor system. The information-processing approach has provided a dominant theoretical and empirical framework for the study of perceptual-motor behavior in general, and for consideration of issues in HCI and human factors in particular. Texts in the area of human factors and HCI (including the present volume) are united in their inclusion of chapters or sections that pertain to the topic of human information processing. Moreover, the design of effective interfaces reflects our knowledge of the perceptual (e.g., visual displays, use of sound, graphics), cognitive (e.g., conceptual models, desktop metaphors), and motoric constraints (e.g., physical design of input devices, ergonomic keyboards) of the human perceptual-motor system.

Technological advances have undoubtedly served to improve the HCI experience. For example, we have progressed beyond the use of computer punch cards and command-line interfaces to more complex tools such as graphical user interfaces, speech recognition, and eye-gaze control systems. As HCI has become not only more effective, but by the same token more elaborate, the importance of the interaction between the various perceptual, cognitive, and motor constraints of the human system has come to the forefront. In our previous chapter, we presented an overview of some topics of research in stimulus-response compatibility in perceptual-motor interactions that we believed were relevant to HCI. In this chapter, we have shifted the focus to current issues in the interaction between attention and action-planning. While action-centered models will require additional development to describe full-body behavior in truly complex environments (e.g., negotiating a busy sidewalk or skating rink), the settings in which these models have been tested are, in fact, very similar to modern HCI environments. Thus, we believe that the relevance of this work for HCI cannot be underestimated. Clearly, considerable research will be necessary to evaluate the applicability of both of these potentially relevant lines of investigation to specific HCI design problems. Nevertheless, the experimental work to date leads us to conclude that the allocation of attention carries an action-centered component. For this reason, an effective interface must be sensitive to the perceptual and action expectations of the user, the specific action associated with a particular response location, the action relationship between that response and those around it, and the degree of translation required to map the perceptual-motor workspaces.

ACKNOWLEDGMENTS

We would like to recognize the financial support of the Natural Sciences and Engineering Research Council of Canada and the Alberta Ingenuity Fund.

References

Abrams, R. A., & Chirst, S. E. (2003). Motion onset captures attention. *Psychological Science, 14*, 427–432.

Abrams, R. A., & Pratt, J. (1996). Spatially-diffuse inhibition affects multiple locations: A reply to Tipper, Weaver, and Watson. *Journal of Experimental Psychology: Human Perception and Performance, 22*, 1294–1298.

Adams, J. A. (1971). A closed-loop theory of motor learning. *Journal of Motor Behavior, 3*, 111–150.

Adams, J. A., & Dijkstra, S. (1966). Short-term memory for motor responses. *Journal of Experimental Psychology, 71*, 314–318.

Allport, A. (1987). Selection for action: Some behavioral and neurophysiological considerations of attention and action. In H. Heuer & A. F. Sanders (Eds.), *Perspectives on perception and action* (pp. 395–419). Hillsdale, NJ: Erlbaum.

Allport, A. (1993). Attention and control: Have we been asking the wrong questions? A critical review of twenty-five years. In D.E. Meyer & S. Kornblum (Eds.), *Attention and performance 14: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (pp. 183–218). Cambridge, MA: MIT Press.

Bayerl, J., Millen, D., & Lewis, S. (1988). Consistent layout of function keys and screen labels speeds user responses. *Proceedings of the Human Factors Society 32nd Annual Meeting* (pp. 334–346). Santa Monica, CA: Human Factors Society.

Bekkering, H. & Neggers, S. F. W. (2002). Visual search is modulated by action intentions. *Psychological Science, 13*, 370–374.

Bekkering, H. & Pratt, J. (2004). Object-based processes in the planning of goal-directed hand movements. *The Quarterly Journal of Experimental Psychology, 57A*, 1345–1368.

Broadbent, D. E. (1958). *Perception and Communication*. New York, NY: Pergamon.

Brooks, V. B. (1974). Some examples of programmed limb movements. *Brain Research, 71*, 299–308.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics, 21*, 601–613.

Castel, A. D., Drummond, E., & Pratt, J. (2005). The effects of action video game experience on the time course of inhibition of return and the efficiency of visual search. *Acta Psychologica, 119*, 217–230.

Castiello, U. (2003). Understanding other people's actions: Intention and attention. *Journal of Experimental Psychology: Human Perception and Performance, 29*, 416–430.

Chapanis, A., & Lindenbaum, L. E. (1959). A reaction time study of four control-display linkages, *Human Factors, 1*, 1–7.

Cheal, M. L., & Lyon, D. R. (1991). Central and peripheral precuing of forced-choice discrimination. *The Quarterly Journal of Experimental Psychology, 43A*, 859–880.

Cherry, E. C. (1953). Some experiments on the recognition speech, with one and with two ears. *Journal of the Acoustical Society of America, 25*, 975–979.

Chua, R., & Elliott, D. (1993). Visual regulation of manual aiming. *Human Movement Science, 12*, 365–401.

Chua, R., Weeks, D. J., & Goodman, D. (2003). Perceptual-motor interaction: Some implications for HCI. In J. A. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, evolving technologies, and emerging applications* (pp. 23–34). Hillsdale, NJ: Lawrence Earlbaum.

Crossman, E. R. F. W., & Goodeve, P. J. (1983). Feedback control of hand movement and Fitts' Law. Paper presented at the meeting of the Experimental Psychology Society, Oxford, July 1963. Published in *Quarterly Journal of Experimental Psychology, 35A*, 251–278.

Davis, G., & Driver, J. (1997). Spreading of visual attention to modally versus amodally completed regions. *Psychological Science, 8*, 275–281.

Deubel, H., & Schneider, W. X. (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research, 36*, 1827–1837.

Deutsch, J. A., & Deutsch, D. (1963). Attention: Some theoretical considerations. *Psychological Review, 70*, 80–90.

Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology: General, 113*, 501–517.

Egly, R., Driver, J., & Rafal, R. D. (1994). Shifting visual attention between objects and locations: Evidence from normal and parietal lesion subjects. *Journal of Experimental Psychology: General, 123*, 161–177.

Elliott, D., Carson, R. G., Goodman, D., & Chua, R. (1991). Discrete vs. continuous visual control of manual aiming. *Human Movement Science, 10*, 393–418.

English, W. K., Engelhart, D. C., & Berman, M. L. (1967). Display-selection techniques for text manipulation. *IEEE Transactions on Human Factors in Electronics, 8*, 5–15.

Eriksen, B. A., & Eriksen, C. W. (1974). Effects of noise letters upon the identification of a target letter in a non-search task. *Perception and Psychophysics, 16*, 143–146.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*, 381–391.

Fitts, P. M. (1964). Perceptual-motor skills learning. In A. W. Melton (Ed.), *Categories of human learning* (pp. 243–285). New York: Academic Press.

Fitts, P. M., & Deninger, R. I. (1954). S-R compatibility: Correspondence among paired elements within stimulus and response codes. *Journal of Experimental Psychology, 48*, 483–491.

Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Belmont, CA: Brooks-Cole.

Fitts, P. M., & Seeger, C. M. (1953). S-R compatibility: Spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology, 46*, 199–210.

Folk, C. L., Remington, R. W., & Johnson, J. C. (1992). Involuntary covert orienting is contingent on attentional control settings. *Journal of Experimental Psychology: Human Perception and Performance, 18*, 1030–1044.

Folk, C. L., Remington, R. W., & Wright, J. H. (1994). The structure of attentional control: Contingent attentional capture by apparent motion, abrupt onset, and color. *Journal of Experimental Psychology: Human Perception and Performance, 20*, 317–329.

Godijn, R., & Theeuwes, J. (2004). The relationship between inhibition of return and saccade trajectory deviations. *Journal of Experimental Psychology: Human Perception and Performance, 30*, 538–554.

Hanna, G. B., Shimi, S. M., & Cuschieri, A. (1998). Task performance in endoscopic surgery is influenced by location of the image display. *Annals of Surgery, 4*, 481–484.

Helsen, W. F., Elliot, D., Starkes, J. L., & Ricker, K. L. (1998). Temporal and spatial coupling of point of gaze and hand movements in aiming. *Journal of Motor Behavior, 30*, 249–259.

Helsen, W. F., Elliott, D., Starkes, J. L., & Ricker, K. L. (2000). Coupling of eye, finger, elbow, and shoulder movements during manual aiming. *Journal of Motor Behavior, 32*, 241–248.

Hick, W. E. (1952). On the rate of gain of information. *The Quarterly Journal of Experimental Psychology, 4*, 11–26.

Hoffman, E. R., & Lim, J. T. A. (1997). Concurrent manual-decision tasks. *Ergonomics, 40*, 293–318.

Howard, L. A., & Tipper, S. P. (1997). Hand deviations away from visual cues: indirect evidence for inhibition. *Experimental Brain Research, 113*, 144–152.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology, 45*, 188–196.

Jonides, J. (1981). Voluntary versus automatic control over the mind's eye's movement. In J. Long & A. Baddley (Eds.), *Attention and performance IX* (pp. 187–203). Hillsdale, NJ: Lawerence Erlbaum.

Kelso, J. A. S. (1982). The process approach to understanding human motor behavior: An introduction. In J. A. S. Kelso (Ed.), *Human Motor Behavior: An Introduction* (pp. 3–19). Hillsdale, NJ: Lawrence Erlbaum.

Kelso, J. A. S. (1995). *Dynamic Patterns: The self-organization of brain and behavior.* Cambridge, MA: MIT Press.

Kelso, J. A. S., Southard, D. L., & Goodman, D. (1979). On the coordination of two-handed movements. *Journal of Experimental Psychology: Human Perception and Performance, 5*, 229–238.

Kveraga, K., Boucher, L., & Hughes, H. C. (2002). Saccades operate in violation of Hick's law. *Experimental Brain Research, 146*, 307–314.

Lyons, J., Elliott, D., Ricker, K. L., Weeks, D. J., & Chua, R. (1999). Action-centered attention in virtual environments. *Canadian Journal of Experimental Psychology, 53*(2), 176–178.

Lyons, J., Weeks, D. J., & Chua, R. (2000a). The influence of object orientation on speed of object identification: Affordance facilitation or cognitive coding? *Journal of Sport & Exercise Psychology, 22*, Supplement, S72.

Lyons, J., Weeks, D. J., & Chua, R. (2000b). Affordance and coding mechanisms in the facilitation of object identification. Paper presented at the Canadian Society for Psychomotor Learning and Sport Psychology Conference, Waterloo, Ontario, Canada.

MacKenzie, C. L., Marteniuk, R. G., Dugas, C., Liske, D., & Eickmeier, B. (1987). Three dimensional movement trajectories in Fitts' task: Implications for control. *The Quarterly Journal of Experimental Psychology, 39A*, 629–647.

Mandryk, R. L., & MacKenzie, C. L. (1999). Superimposing display space on workspace in the context of endoscopic surgery. *ACM CHI Companion,* 284–285.

Marteniuk, R. G. (1976). *Information processing in motor skills*. New York: Holt, Rinehart and Winston.

Marteniuk, R. G., MacKenzie, C. L., & Leavitt, J. L. (1988). Representational and physical accounts of motor control and learning: Can they account for the data? In A. M. Colley & J. R. Beech (Eds.), *Cognition and action in skilled behavior* (pp. 173–190). Amsterdam: Elsevier Science Publishers.

Marteniuk, R. G., MacKenzie, C. L., Jeannerod, M., Athenes, S., & Dugas, C. (1987). Constraints on human arm movement trajectories. *Canadian Journal of Psychology, 41,* 365–378.

Maylor, E. A., & Hockey, R. (1985). Inhibitory component of externally controlled covert orienting in visual space. *Journal of Experimental Psychology: Human Perception and Performance, 11*, 777–787.

Meegan, D. V., & Tipper, S. P. (1998). Reaching into cluttered visual environments: Spatial and temporal influences of distracting objects. *The Quarterly Journal of Experimental Psychology, 51A*, 225–249.

Meegan, D. V., & Tipper, S. P. (1999). Visual search and target-directed action. *Journal of Experimental Psychology: Human Perception and Performance, 25*, 1347–1362.

Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Smith, J. E. K. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review, 95*, 340–370.

Müller, H. J., & Rabbitt, P. M. A. (1989). Reflexive and voluntary orienting of visual attention: Time course of activation and resistance to interruption. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 315–330.

Nicoletti, R., & Umiltà, C. (1989). Splitting visual space with attention. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 164–169.

Nicoletti, R., & Umiltà, C. (1994). Attention shifts produce spatial stimulus codes. *Psychological Research, 56*, 144–150.

Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116*, 220–244.

Patterson, J. T., & Lee, T. D. (2005, in press). Learning a new human-computer alphabet: The role of similarity and practice. *Acta Psychologica*.

Payne, S. J. (1995). Naïve judgments of stimulus-response compatibility. *Human Factors, 37*, 495–506.

Posner, M. I. (1982). Cumulative development of attentional theory. *American Psychologist, 37*, 168–179.

Posner, M. I., & Cohen, Y. (1984). Components of visual orienting. In H. Bouma & D. G. Bouwhuis (Eds.), *Attention and Performance X* (pp. 531–556). Hillsdale, NJ: Lawrence Earlbaum.

Posner, M. I., & Keele, S. W. (1969). Attentional demands of movement. *Proceedings of the 16th Congress of Applied Psychology*. Amsterdam: Swets and Zeitlinger.

Posner, M. I., Nissen, M. J., & Ogden, W. C. (1978). Attended and unattended processing modes: The role of set for spatial location. In H. Pick & E. Saltzman (Eds.), *Modes of perceiving and processing information* (pp. 137–157). Hillsdale, NJ: Lawrence Earlbaum.

Pratt, J., & Sekuler, A. B. (2001). The effects of occlusion and past experience on the allocation of object-based attention. *Psychonomic Bulletin & Review, 8*, 721–727.

Pratt, J., & Abrams, R. A. (1994). Action-centered inhibition: Effects of distractors in movement planning and execution. *Human Movement Science, 13*, 245–254.

Pratt, J., & McAuliffe, J. (2001). The effects of onsets and offsets on visual attention. *Psychological Research, 65*(3), 185–191.

Pratt, J., & Sekuler, A. B. (2001). The effects of occlusion and past experience on the allocation of object-based attention. *Psychonomic Bulletin and Review, 8*(4), 721–727.

Proctor, R. W., & Vu, K. L. (2003). Human information processing. In J. A. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, evolving technologies, and emerging applications* (pp. 35–50). Hillsdale, NJ: Lawrence Earlbaum.

Proctor, R. W., & Lu, C. H. (1994). Referential coding and attention-shifting accounts of the Simon effect. *Psychological Research, 56*, 185–195.

Proctor, R. W., & Van Zandt, T. (1994). *Human factors in simple and complex systems*. Boston: Allyn and Bacon.

Rafal, R. D., Calabresi, P. A., Brennan, C. W., & Sciolto, T. K. (1989). Saccade preparation inhibits reorienting to recently attended locations. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 673–685.

Rizzolatti, G., Riggio, L., & Sheliga, B. M. (1994). Space and selective attention. In C. Umiltà & M. Moscovitch (Eds.), *Attention and performance XV* (pp. 231–265). Cambridge: MIT Press.

Rizzolatti, G., Riggio, L., Dascola, J., & Umilta, C. (1987). Reorienting attention across the horizontal and vertical meridians: Evidence in favor of a premotor theory of attention. *Neuropsychologia, 25*, 31–40.

Rubichi, S., Nicoletti, R, Iani, C., & Umiltà, C. (1997). The Simon effect occurs relative to the direction of an attention shift. *Journal of Experimental Psychology: Human Perception and Performance, 23,* 1353–1364.

Rutkowski, C. (1982). An introduction to the human applications standard computer interface, part I: Theory and principles. *Byte, 7*, 291–310.

Shepherd, M., Findlay, J. M., & Hockey, R. J. (1986) The relationship between eye-movements and spatial attention. *The Quarterly Journal of Experimental Psychology, 38A*, 475–491.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer, 16*, 57–69.

Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction.* Reading, MA: Addison-Wesley Publishing Company.

Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. *CHI Letters, 2*, 281–288.

Simon, J. R. (1968). Effect of ear stimulated on reaction time and movement time. *Journal of Experimental Psychology, 78*, 344–346.

Simon, J. R., & Rudell, A. P. (1967). Auditory S-R compatibility: The effect of an irrelevant cue on information processing. *Journal of Applied Psychology, 51*, 300–304.

Spence, C., Lloyd, D., McGlone, F., Nichols, M. E. R., & Driver, J. (2000). Inhibition of return is supramodal: A demonstration between all possible pairings of vision, touch, and audition. *Experimental Brain Research, 134*, 42–48.

Stelmach, G. E. (1982). Information-processing framework for understanding human motor behavior. In J. A. S. Kelso (Ed.), *Human Motor Behavior: An Introduction* (pp. 63–91). Hillsdale, NJ: Lawrence Erlbaum.

Stoffer, T. H. (1991). Attentional focusing and spatial stimulus response compatibility. *Psychological Research, 53*, 127–135.

Stoffer, T. H., & Umiltà, C. (1997). Spatial stimulus coding and the focus of attention in S-R compatibility and the Simon effect. In B. Hommel & W. Prinz (eds.), *Theoretical issues in stimulus-response compatibility* (pp. 373–398). Amsterdam: Elsevier Science B.V.

Taylor, T. L., & Klein, R. M. (2000). Visual and motor effects in inhibition of return. *Journal of Experimental Psychology: Human Perception and Performance, 26*, 1639–1656.

Telford, C. W. (1931). The refractory phase of voluntary and associative responses. *Journal of Experimental Psychology, 14*, 1–36.

Tenner, E. (1996). *Why things bite back: Technology and the revenge of unintended consequences*. NY: Alfred A. Knopf.

Tipper, S. P., Howard, L. A., & Houghton, G. (1999). Behavioral consequences of selection form neural population codes. In S. Monsell & J. Driver (Eds.), *Attention and performance XVIII* (pp. 223–245). Cambridge, MA: MIT Press.

Tipper, S. P., Howard, L. A., & Jackson, S. R. (1997). Selective reaching to grasp: Evidence for distractor interference effects. *Visual Cognition, 4*, 1–38.

Tipper, S. P., Lortie, C., & Baylis, G. C. (1992). Selective reaching evidence for action-centered attention. *Journal of Experimental Psychology: Human Perception and Performance, 18*, 891–905.

Tipper, S. P., Meegan, D., & Howard, L. A. (2002). Action-centered negative priming: Evidence for reactive inhibition. *Visual Cognition, 9*, 591–614.

Tipper, S. P., Weaver, B., & Watson, F. L. (1996). Inhibition of return to successively cued spatial locations: Commentary on Pratt and Abrams (1995). *Journal of Experimental Psychology: Human Perception and Performance, 22*, 1289–1293.

Treisman, A. M. (1964a). The effect of irrelevant material on the efficiency of selective listening. *American Journal of Psychology, 77*, 533–546.

Treisman, A. M. (1964b). Verbal cues, language, and meaning in selective attention. *American Journal of Psychology, 77*, 206–219.

Treisman, A. M. (1986). Features and objects in visual processing. *Scientific American, 255*, 114–125.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12*, 97–136.

Tremblay, L., Welsh, T. N., & Elliott, D. (2005). Between-trial inhibition and facilitation in goal-directed aiming: Manual and spatial asymmetries. *Experimental Brain Research, 160*, 79–88.

Tucker, M., & Ellis, R. (1998). On the relations between seen objects and components of potential actions. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 830–846.

Umiltà, C., & Nicoletti, R. (1992). An integrated model of the Simon effect. In J. Alegria, D. Holender, J. Junca de Morais, & M. Radeau (Eds.), *Analytic approaches to human cognition* (pp. 331–350). Amsterdam: North-Holland.

Vu, K-P. L., & Proctor, R. W. (2003). Naïve and experienced judgments of stimulus-response compatibility: Implications for interface design. *Ergonomics, 46*, 169–187.

Wallace, R. J. (1971). S-R compatibility and the idea of a response code. *Journal of Experimental Psychology, 88*, 354–360.

Weir, P. L., Weeks, D. J., Welsh, T. N., Elliott, D., Chua, R., Roy, E. A., & Lyons, J. (2003). Action-centered distractor effects in discrete control selection. *Experimental Brain Research, 149*, 207–213.

Welford, A. T. (1968). *Fundamentals of skill.* London: Methuen.

Welsh, T. N., & Elliott, D. (2004a). Movement trajectories in the presence of a distracting stimulus: Evidence for a response activation model of selective reaching. *Quarterly Journal of Experimental Psychology–Section A, 57*, 1031–1057.

Welsh, T. N., & Elliott, D. (2004b). The effects of response priming and inhibition on movement planning and execution. *Journal of Motor Behavior, 36*, 200–211.

Welsh, T. N., & Elliott, D. (2005). The effects of response priming on the planning and execution of goal-directed movements in the presence of a distracting stimulus. *Acta Psychologica, 119*, 123–142.

Welsh, T. N., Elliott, D., Anson, J. G., Dhillon, V., Weeks, D. J., Lyons, J. L., & Chua, R. (2005). Does Joe influence Fred's action? Inhibition of return across different nervous systems. *Neuroscience Letters, 385*, 99–104.

Welsh, T. N., Elliott, D., & Weeks, D. J. (1999). Hand deviations towards distractors: Evidence for response competition. *Experimental Brain Research, 127*, 207–212.

Welsh, T. N., Lyons, J., Weeks, D. J., Anson, J. G., Chua, R., Mendoza, J. E., & Elliott, D. (in press). Within- and between-nervous system inhibition of return: Observation is as good as performance. *Psychonomic Bulletin and Review*.

Welsh, T. N., & Pratt, J. (2005, November). The salience of offset and onset stimuli to attention and motor systems: Evidence through distractor interference. Paper presented at the annual meeting of the Canadian Society for Psychomotor Learning and Sports Psychology, Niagara Falls, ON, Canada.

Wright, R. D., & Richard, C. M. (1996). Inhibition of return at multiple locations in visual space. *Canadian Journal of Experimental Psychology, 50*, 324–327.

◆2◆

HUMAN INFORMATION PROCESSING: AN OVERVIEW FOR HUMAN-COMPUTER INTERACTION

Robert W. Proctor
Purdue University

Kim-Phuong L. Vu
California State University Long Beach

Human Information Processing Approach **44**
Information-processing Methods **45**
Signal Detection Methods and Theory 45
Chronometric Methods 45
Speed-Accuracy Methods 46
Psychophysiological and Neuroimaging Methods 46
Information-Processing Models **47**
Discrete and Continuous Stage Models 47
Sequential Sampling Models 48
Information Processing in Choice Reaction Tasks . . . **48**
Stimulus Identification 48
Response Selection 48
Response Execution 50
Memory in Information Processing **51**
Short-Term (Working) Memory 51
Long-Term Memory 52
Other Factors Affecting Retrieval of Earlier Events 53
Attention in Information Processing **54**
Models of Attention 54
Automaticity and Practice 56
Problem Solving and Decision Making **56**
New Developments and Alternative Approaches **57**
Cognitive Neuroscience 57
The Ecological Approach 57
Cybernetic Approach 58
Situated Cognition 58
Summary and Conclusion **58**
References **58**

It is natural for an applied psychology of human-computer interaction to be based theoretically on information-processing psychology.
—Card, Moran, & Newell, 1983

Human-computer interaction (HCI) is fundamentally an information-processing task. In interacting with a computer, a user has specific goals and subgoals in mind. The user initiates the interaction by giving the computer commands that are directed toward accomplishing those goals. The commands may activate software programs designed to allow specific types of tasks, such as word processing or statistical analysis to be performed. The resulting computer output, typically displayed on a screen, must provide adequate information for the user to complete the next step, or the user must enter another command to obtain the desired output from the computer. The sequence of interactions to accomplish the goals may be long and complex, and several alternative sequences, differing in efficiency, may be used to achieve these goals. During the interaction, the user is required to identify displayed information, select responses based on the displayed information, and execute those responses. The user must search the displayed information and attend to the appropriate aspects of it. She or he must also recall the commands and resulting consequences of those commands for different programs, remember information specific to the task that is being performed, and make decisions and solve problems during the process. For the interaction between the computer and user to be efficient, the interface must be designed in accordance with the user's information processing capabilities.

HUMAN INFORMATION PROCESSING APPROACH

The rise of the human information-processing approach in psychology is closely coupled with the growth of the fields of cognitive psychology, human factors, and human engineering. Although research that can be classified as falling within these fields has been conducted since the last half of the 19th century, their formalization dates back to World War II (see Hoffman & Deffenbacher, 1992). As part of the war efforts, experimental psychologists worked along with engineers on applications associated with using the sophisticated equipment being developed. As a consequence, the psychologists were exposed not only to applied problems but also to the techniques and views being developed in areas such as communications engineering (see Roscoe, 2005). Many of the concepts from engineering (for instance, the notion of transmission of information through a limited capacity communications channel) were seen as applicable to analyses of human performance.

The human information-processing approach is based on the idea that human performance, from displayed information to response, is a function of several processing stages. The nature of these stages, how they are arranged, and the factors that influence how quickly and accurately a particular stage operates can be discovered through appropriate research methods. It is often said that the central metaphor of the information-processing approach is that a human is like a computer (e.g., Lachman, Lachman, & Butterfield, 1979). However, even more fundamental than the computer metaphor is the assumption that the human is a complex system that can be analyzed in terms of subsystems and their interrelation. This point is evident in the work of researchers on attention and performance, such as Paul Fitts (1951) and Donald Broadbent (1958), who were among the first to adopt the information-processing approach in the 1950s.

The systems perspective underlies not only human information-processing but also human factors and HCI, providing a direct link between the basic and applied fields (Proctor & Van Zandt, 1994). Human factors in general, and HCI in particular, begin with the fundamental assumption that a human-machine system can be decomposed into machine and human subsystems, each of which can be analyzed further. The human information-processing approach provides the concepts, methods, and theories for analyzing the processes involved in the human subsystem. Posner (1986) stated, "Indeed, much of the impetus for the development of this kind of empirical study stemmed from the desire to integrate description of the human within overall systems" (p. V-6). Young, Clegg, and Smith (2004) emphasized that the most basic distinction between three processing stages (perception, cognition, and action), as captured in a block diagram model of human information processing, is important even for understanding the dynamic interactions of an operator with a vehicle for purposes of computer-aided augmented cognition. They noted:

> This block diagram model of the human is important because it not only models the flow of information and commands between the vehicle and the human, it also enables access to the internal state of the human at various parts of the process. This allows the modeling of what a cognitive measurement system might have access to (internal to the human), and how that measurement might then be used as part of a closed-loop human-machine interface system. (pp. 261–262)

In the first half of the 20th century, the behaviorist approach predominated in psychology, particularly in the United States. Within this approach, many sophisticated theories of learning and behavior were developed that differed in various details (Bower & Hilgard, 1981). However, the research and theories of the behaviorist approach tended to minimize the role of cognitive processes and were of limited value to the applied problems encountered in World War II. The information-processing approach was adopted because it provided a way to examine topics of basic and applied concern, such as attention, that were relatively neglected during the behaviorist period. It continues to be the main approach in psychology, although contributions have been made from other approaches, some of which we will consider in the last section of the chapter.

Within HCI, human information-processing analyses are used in two ways. First, empirical studies evaluate the information-processing requirements of various tasks in which a human uses a computer. Second, computational models are developed which are intended to characterize human information processing when interacting with computers, and predict human performance with alternative interfaces. In this chapter, we survey methods used to study human information processing and summarize the major findings and the theoretical frameworks developed to explain them. We also tie the methods, findings, and theories to HCI issues to illustrate their use.

INFORMATION-PROCESSING METHODS

Any theoretical approach makes certain presuppositions and tends to favor some methods and techniques over others. Information-processing researchers have used behavioral and, to an increasing extent, psychophysiological measures, with an emphasis on chronometric (time-based) methods. There also has been a reliance on flow models that are often quantified through computer simulation or mathematical modeling.

Signal Detection Methods and Theory

One of the most useful methods for studying human information processing is that of signal detection (Macmillan & Creelman, 2005). In a signal-detection task, some event is classified as a signal and the subject's task is to detect whether the signal is present. Trials in which it is not present are called "noise trials." The proportion of trials in which the signal is correctly identified as present is called the "hit rate," and the proportion of trials in which the signal is incorrectly identified as present is called the "false alarm rate." By using the hit and false alarm rates, whether the effect of a variable is on detectability or response bias can be evaluated.

Signal-detection theory is often used as the basis for analyzing data from such tasks. This theory assumes that the response on each trial is a function of two discrete operations: encoding and decision. In a trial, the subject samples the information presented and decides whether this information is sufficient to warrant a *signal present* response. The sample of information is assumed to provide a value along a continuum of evidence states regarding the likelihood that the signal was present. The noise trials form a probability distribution of states, as do the signal trials. The decision that must be made on each trial can be characterized as to whether the event is from the signal or noise distribution. The subject is presumed to adopt a criterion value of evidence above which he or she responds "signal present" and below which he or she responds "signal absent."

In the simplest form, the distributions are assumed to be normal and equal variance. In this case, a measure of detectability (d') can be derived. This measure represents the difference in the means for the signal and noise distributions in standard deviation units. A measure of response bias (β), which represents the relative heights of the signal and noise distributions at the criterion, can also be calculated. This measure reflects the subject's overall willingness to say "signal present," regardless of whether it actually is present. There are numerous alternative measures of detectability and bias based on different assumptions and theories, and many task variations to which they can be applied (see Macmillan & Creelman, 2005).

Signal-detection analyses have been particularly useful because they can be applied to any task that can be depicted in terms of binary discriminations. For example, the proportion of words in a memory task correctly classified as old can be treated as a hit rate and the proportion of new lures classified as old can be treated as a false alarm rate (Lockhart & Murdock, 1970). In cases such as these, the resulting analysis helps researchers determine whether variables are affecting detectability of an item as old or response bias.

An area of research in which signal-detection methods have been widely used is that of vigilance (Parasuraman & Davies, 1977). In a typical vigilance task, a display is monitored for certain changes in it (e.g., the occurrence of an infrequent stimulus). Vigilance tasks are common in the military, but many aspects also can be found in computer-related tasks such as monitoring computer network operations (Percival & Noonan, 1987). A customary finding for vigilance tasks is the vigilance decrement, in which the hit rate decreases as time on the task increases. The classic example of this vigilance decrement is that, during World War II, British radar observers detected fewer of the enemy's radar signals after 30 minutes in a radar observation shift (Mackworth, 1948). Parasuraman and Davies concluded that, for many situations, the primary cause of the vigilance decrement is an increasingly strict response criterion. That is, the false alarm rate as well as the hit rate decreases as a function of time on task. They provided evidence that perceptual sensitivity also seems to decrease when the task requires comparison of each event to a standard held in memory and the event rate is high. Subsequently, See, Howe, Warm, and Dember (1995) concluded that a decrease in perceptual sensitivity occurs for a broad range of tasks. Although signal-detection theory can be used to help determine whether a variable affects encoding quality or decision, as in the vigilance example, it is important to keep in mind that the measures of detectability and bias are based on certain theoretical assumptions. Balakrishnan (1998) argued, on the basis of an analysis that does not require the assumptions of signal-detection theory, that the vigilance decrement is not a result of a biased placement of the response criterion, even when the signal occurs infrequently and time on task increases.

Chronometric Methods

Chronometric methods, for which time is a factor, have been the most widely used for studying human information processing. Indeed, Lachman et al. (1979) portrayed reaction time (RT) as the main dependent measure of the information-processing approach. Although many other measures are used, RT still predominates in part because of its sensitivity and in part because of the sophisticated techniques that have been developed for analyzing RT data.

A technique called the "subtractive method," introduced by F. C. Donders (1868/1969) in the 1860s, was revived in the 1950s and 60s. This method provides a way to estimate the duration of a particular processing stage. The assumption of the subtractive method is that a series of discrete processing stages intervene between stimulus presentation and response execution. Through careful selection of tasks that differ by a single stage, the RT for the easier task can be subtracted from that for the more difficult task to yield the time for the additional process. The subtractive method has been used to estimate the durations of a variety of processes, including rates of mental rotation (approximately 12 to 20 ms per degree of rotation; Shepard & Metzler, 1971) and memory search (approximately 40 ms per item; Sternberg, 1969). An application of the subtractive method to HCI would be, for example, to compare the time to find a target link on two web pages that are identical except for the number of links displayed, and to attribute the extra time

to the additional visual search required for the more complex web page.

The subtractive method has several limitations (Pachella, 1974). First, it is only applicable when discrete, serial processing stages can be assumed. Second, the processing for the two tasks being compared must be equivalent except for the additional process that is being evaluated. This requires an assumption of pure insertion, which is that the additional process for the more complex of two tasks can be inserted without affecting the processes held in common by the two tasks. However, this assumption often is not justified.

Sternberg (1969) developed the additive factors method to allow determination of the processes involved in performing a task. The additive factors method avoids the problem of pure insertion because the crucial data are whether two variables affect RT for the same task in an additive or interactive manner. Sternberg assumed, as did Donders, that information processing occurs in a sequence of discrete stages, each of which produces a constant output that serves as input to the next stage in the sequence. With these assumptions, he showed that two variables that affect different stages should have additive effects on RT. In contrast, two variables that affect the same stage should have interactive effects on RT. Sternberg performed detailed analyses of memory search tasks in which a person holds a set of letters or digits in memory and responds to a target stimulus by indicating whether it is in the memory set. Based on the patterns of additive and interactive effects that he observed, Sternberg concluded that the processing in such tasks involves four stages: target identification, memory search, response selection, and response execution. Grobelny, Karwowski, and Drury (2005) provided an application of additive-factors logic to usability of graphical icons in the design of HCI interfaces. Mode of icon array (menu or dialog box), number of icons, and difficulty of movement had additive effects on response times, implying that these variables affect different processing stages.

Both the subtractive and additive factors methods have been challenged on several grounds (Pachella, 1974). First, the assumption of discrete serial stages with constant output is difficult to justify in many situations. Second, both methods rely on analyses of RT, without consideration of error rates. This can be problematic because performance is typically not error free, and, as described below, speed can be traded for accuracy. Despite these limitations, the methods have proved to be robust and useful (Sanders, 1998). For example, Salthouse (2005) noted that the process analysis approach employed in contemporary research into aging effects on cognitive abilities "has used a variety of analytical methods such as subtraction, additive factors . . . to partition the variance in the target variable into theoretically distinct processes" (p. 288).

Speed-Accuracy Methods

The function relating response speed to accuracy is called the "speed-accuracy tradeoff" (Pachella, 1974). The function, illustrated in Fig. 2.1, shows that very fast responses can be performed with chance accuracy, and accuracy will increase as responding slows down. Of importance is the fact that when accuracy is high, as in most RT studies, a small increase in errors can result in a large decrease in RT. With respect to text entry on computing devices, MacKenzie and Soukoreff (2002) stated, "Clearly, both speed and accuracy must be measured and analyzed. . . . Participants can enter text more quickly if they are willing to sacrifice accuracy" (pp. 159–160).

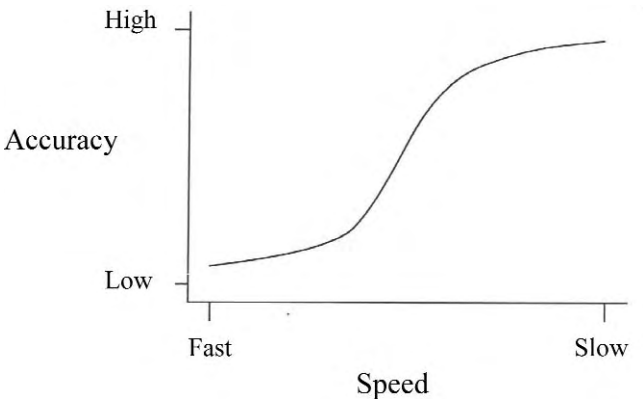


FIGURE 2.1. Speed-accuracy operating characteristic curve. Faster responding occurs at the cost of lower accuracy.

In speed-accuracy tradeoff studies, the speed-accuracy criterion is varied between blocks of trials or between subjects by using different instructions regarding the relative importance of speed versus accuracy, varying payoffs such that speed or accuracy is weighted more heavily, or imposing different response deadlines (Wickelgren, 1977). These studies have the potential to be more informative than RT studies because they can provide information about whether variables affect the intercept (time at which accuracy exceeds chance), asymptote (the maximal accuracy), and rate of ascension from the intercept to the asymptote, each of which may reflect different processes. For example, Boldini, Russo, and Avons (2004) obtained evidence favoring dual-process models of recognition memory over single-process models by varying the delay between a visually presented test word and a signal to respond. Recognition accuracy benefited from a modality match at study and test (better performance when the study words were also visual rather than auditory) at short response-signal delays, but it benefited from deep processing during study (judging pleasantness) over shallow processing (repeating aloud each word) at long response-signal delays. Boldini et al. interpreted these results as consistent with the view that recognition judgments are based on a fast familiarity process or a slower recollection process.

Because the speed-accuracy criterion is manipulated in addition to any other variables of interest, much more data must be collected in a speed-accuracy study than in a RT study. Consequently, use of speed-accuracy methods has been restricted to situations in which the speed-accuracy relation is of major concern, rather than being widely adopted as the method of choice.

Psychophysiological and Neuroimaging Methods

In recent years, psychophysiological methods have been used to evaluate implications of information-processing models and to relate the models to brain processes. Such methods can pro-

vide details regarding the nature of processing by examining physiological activity as a task is being performed. The most widely used method involves measurement of electroencephalograms (EEGs), which are recordings of changes in brain activity as a function of time measured by electrodes placed on the scalp (Rugg & Coles, 1995). Of most concern for information-processing research are event-related potentials (ERPs), which are the changes in brain activity that are elicited by an event such as stimulus presentation or response initiation. ERPs are obtained by averaging across many trials of a task to remove background EEG noise and are thought to reflect postsynaptic potentials in the brain.

There are several features of the ERP that represent different aspects of processing. These features are labeled according to their polarity, positive (P) or negative (N), and their sequence or latency. The first positive (P1) and first negative (N1) components are associated with early perceptual processes. They are called "exogenous components" because they occur in close temporal proximity to the stimulus event and have a stable latency with respect to it. Later components reflect cognitive processes and are called "endogenous" because they are a function of the task demands and have a more variable latency than the exogenous components. One such component that has been studied extensively is the P3 (or P300), which represents post-perceptual processes. When an occasional target stimulus is interspersed in a stream of standards, the P3 is observed in response to targets, but not to standards. By comparing the effects of task manipulations on various ERP components such as P3, their onset latencies, and their scalp distributions, relatively detailed inferences about the cognitive processes can be made.

One example of applying a P3 analysis to HCI is a study by Trimmel and Huber (1998). In their study, subjects performed three HCI tasks (text editing, programming, and playing the game *Tetris*) for seven minutes each. They also performed comparable paper/pencil tasks in three other conditions. The P3 was measured after each experimental task by having subjects monitor a stream of high- and low-pitched tones, keeping count of each separately. The P3 varied as a function of type of task, as well as medium (computer vs. paper/pencil). The amplitude of the P3 was smaller following the HCI tasks than following the paper/pencil tasks, suggesting that the HCI tasks caused more fatigue or depletion of cognitive resources than the paper/pencil task. The P3 latency was shorter after the programming task than after the others, which the authors interpreted as an after-effect of highly focused attention.

Another measure that has been used in studies of human information processing is the lateralized readiness potential (LRP; Eimer, 1998). The LRP can be recorded in choice-reaction tasks that require a response with the left or right hand. It is a measure of differential activation of the lateral motor areas of the visual cortex that occurs shortly before and during execution of a response. The asymmetric activation favors the motor area contralateral to the hand making the response, because this is the area that controls the hand. The LRP has been obtained in situations in which no overt response is ever executed, allowing it to be used as an index of covert, partial-response activation. The LRP is thus a measure of the difference in activity from the two sides of the brain that can be used as an indicator of covert reaction tendencies, to determine whether a response has been prepared even when it is not actually executed. It can also be used to determine whether the effects of a variable are prior or subsequent to response preparation.

Electrophysiological measurements do not have the spatial resolution needed to provide precise information about the brain structures that produce the recorded activity. Recently developed neuroimaging methods, including positron-emission tomography (PET) and functional magnetic resonance imaging (fMRI), measure changes in blood flow associated with neuronal activity in different regions of the brain (Huettel, Song, & McCarthy, 2004). These methods have poor temporal resolution but much higher spatial resolution than the electrophysiological methods. In an imaging study, both control and experimental tasks are performed, and the functional neuroanatomy of the cognitive processes is derived by subtracting the image during the control task from that during the experimental task.

Application of cognitive neuroscience to HCI has been advocated under the heading of neuroergonomics. According to Parasuraman (2003), "Neuroergonomics focuses on investigations of the neural bases of mental functions and physical performance in relation to technology, work, leisure, transportation, health care and other settings in the real world" (p. 5). Neuroergonomics has the goal of using knowledge of the relation between brain function and human performance to design interfaces and computerized systems that are sensitive to brain function with the intent of increasing the efficiency and safety of human-machine systems.

INFORMATION-PROCESSING MODELS

Discrete and Continuous Stage Models

It is common to assume that the processing between stimuli and responses consists of a series of discrete stages for which the output for one stage serves as the input for the next, as Donders and Sternberg assumed. This assumption is made for the Model Human Processor (Card et al., 1983) and the Executive-Process Interactive Control (EPIC; Meyer & Kieras, 1997) architectures, both of which have been applied to HCI. However, models can be developed that allow for successive processing stages to operate concurrently. A well-known model of this type is McClelland's (1979) cascade model, in which partial information at one subprocess, or stage, is transferred to the next. Each stage is continuously active, and its output is a continuous value that is always available to the next stage. The final stage results in selection of which of the possible alternative responses to execute. Many parallel distributed processing, or neural network, models are of a continuous nature.

According to J. Miller (1988), models of human information processing can be classified as discrete or continuous along three dimensions: (a) representation, (b) transformation, and (c) transmission. "Representation" refers to whether the input and output codes for the processing stage are continuous or discrete. "Transformation" refers to whether the operation performed by the processing stage (e.g., spatial transformation) is continuous or discrete. "Transmission" is classified as discrete if the processing of successive stages does not overlap temporally.

The discrete stage model proposed by Sternberg (1969) has discrete representation and transmission, whereas the cascade model proposed by McClelland (1979) has continuous representation, transmission, and transformation. Models can be intermediate to these two extremes. For example, J. Miller's (1988) asynchronous discrete coding model assumes that most stimuli are composed of features, and these features are identified separately. Discrete processing occurs for feature identification, but once a feature is identified, this information can be passed to response selection while the other features are still being identified.

Sequential Sampling Models

Sequential sampling models are able to account for both RT and accuracy, and consequently, the tradeoff between them (Ratcliff & Smith, 2004; Van Zandt, Colonius, & Proctor, 2000). According to such models, information from the stimulus is sequentially sampled, resulting in a gradual accumulation of information on which selection of one of the alternative responses is based. A response is selected when the accumulated information exceeds a threshold amount required for that response. Factors that influence the quality of information processing have their effects on the rate at which the information accumulates, whereas factors that bias speed-versus-accuracy or specific responses have their effects on the response thresholds.

Balakrishnan (1998) argued that sequential sampling may be a factor even when the experiment does not stress speed of responding. As mentioned before, he showed that an analysis of vigilance data that does not make the assumptions of signal-detection theory suggests that attribution of the vigilance decrement to a change toward a more conservative response bias is incorrect. One reason why signal detection theory may lead to an incorrect conclusion is that the model assumes that the decision is based on a fixed sample of information, rather than information that is accumulating across time. Balakrishnan argued that even though there are no incentives to respond quickly in the typical vigilance task, subjects may choose not to wait until all of the stimulus information has been processed before responding. He proposed that a sequential sampling model, in which the subject continues to process the information until a stopping rule condition is satisfied, provides a better depiction. In this model, there are two potential sources of bias: the stopping rule and decision rule. Based on this model, Balakrishnan concluded that there is a response bias initially when the signal rate is low, and that the vigilance decrement is due to a gradual reduction of this response bias toward a more optimal decision during the time-course of the vigil.

INFORMATION PROCESSING IN CHOICE REACTION TASKS

In a typical choice reaction task in which each stimulus is assigned to a unique response, it is customary to distinguish between three stages of processing: stimulus identification, response selection, and response execution (Proctor & Van Zandt, 1994). The stimulus identification stage involves processes that are entirely dependent on stimulus properties. The response-selection stage concerns those processes involved in determining which response to make to each stimulus. Response execution refers to programming and execution of motor responses. Based on additive factors logic, Sanders (1998) decomposed the stimulus identification stage into three subcategories and the response execution stage into two subcategories, resulting in six stages (see Fig. 2.2).

Stimulus Identification

The preprocessing stage of stimulus identification refers to peripheral sensory processes involved in the conduction of the sensory signal along the afferent pathways to the sensory projection areas of the cerebral cortex. These processes are affected by variables such as stimulus contrast and retinal location. As stimulus contrast, or intensity, increases, RT decreases until reaching asymptote. For example, Bonin-Guillaume, Possamäi, Blin, and Hasbroucq (2000) had younger and older adults perform a two-choice reaction task, in which a left or right keypress was made to a bright or dim light positioned to the left or right. Stimulus intensity interacted with age, with RTs being about 25 ms shorter to a bright stimulus than to a dim stimulus for younger adults compared to 50 ms for older adults. The effect of stimulus intensity did not interact with variables that affect response selection or motor adjustment, suggesting that the age-related deficit in sensory preprocessing did not affect the later processing stages.

Feature extraction involves lower-level perceptual processing based in area V1 (the visual cortex) and other early visual cortical areas. Stimulus discriminability, word priming, and stimulus quality affect the feature extraction process. For example, manipulations of stimulus quality (such as superimposing a grid) slow RT, presumably by creating difficulty for the extraction of features. Identification itself is influenced by word frequency and mental rotation. The latter refers to the idea that, when a stimulus is rotated from the upright position, the time it takes to identify the stimulus increases as an approximately linear function of angular deviation from upright (Shepard & Metzler, 1971). This increase in identification time is presumed to reflect a normalization process by which the image is mentally rotated in a continuous manner to the upright position.

Response Selection

Response selection refers to those processes involved in determining what response to make to a particular stimulus. It is affected by the number of alternatives, stimulus-response compatibility, and precuing. RT increases as a logarithmic function of the number of stimulus-response alternatives (Hick, 1952; Hyman 1953). This relation is known as the Hick-Hyman law, which for N equally likely alternatives is:

$$RT = a + b \log_2 N$$

where a is the base processing time and b is the amount that RT increases with increases in N. The slope of the Hick-Hyman

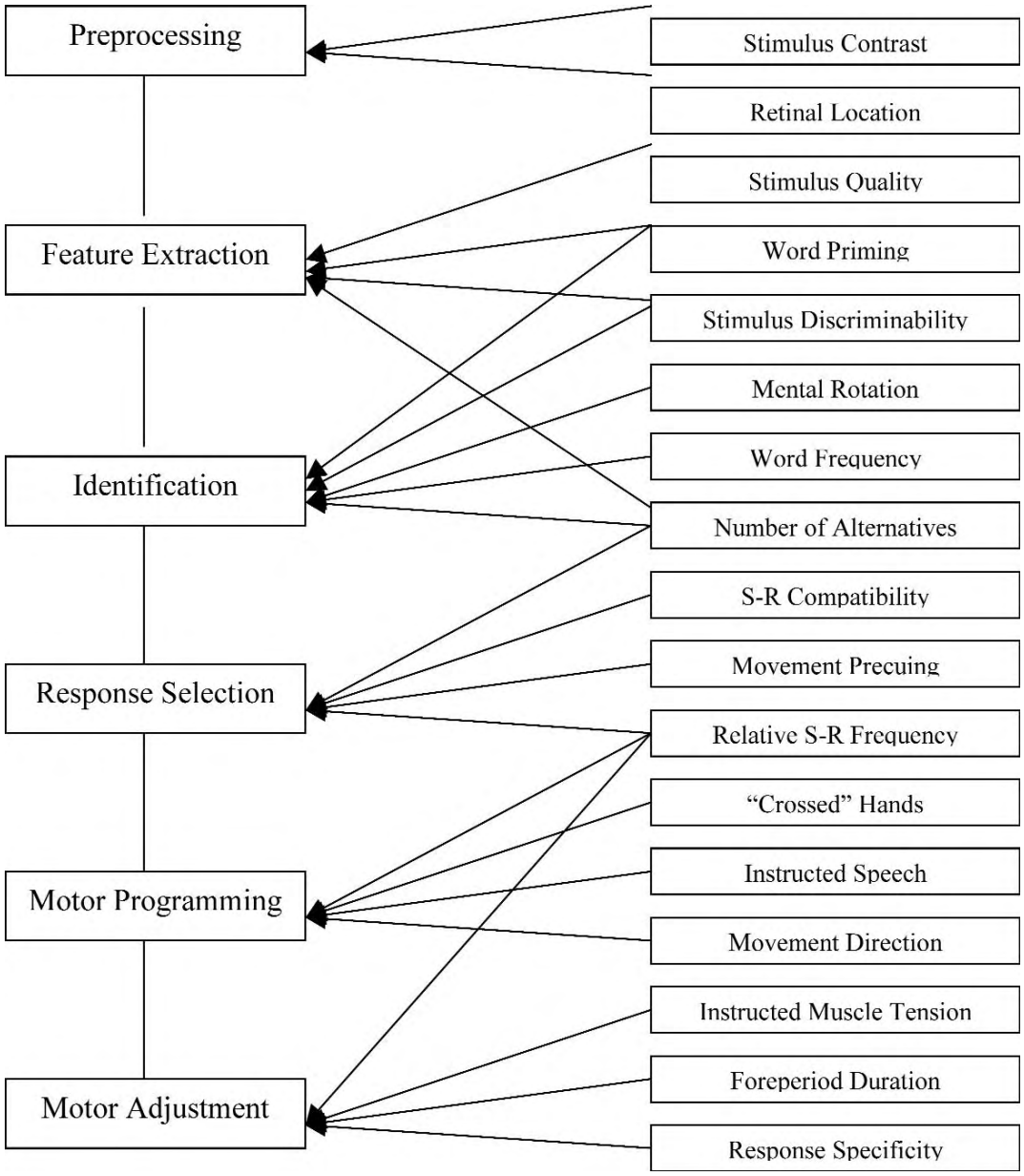


FIGURE 2.2. Information-processing stages and variables that affect them, based on Sanders' (1998) taxonomy.

function is influenced by many factors. For example, the slope decreases as subjects become practiced at a task (Teichner & Krebs, 1974). Usher, Olami, and McClelland (2002) provided evidence from fits of a sequential sampling model that the Hick-Hyman law is due to subjects' adjusting their response criteria upward as the number of alternatives increases, in an attempt to maintain a constant high level of accuracy.

One variable that influences the slope of the Hick-Hyman function is stimulus-response compatibility, which has considerable impact on response-selection efficiency (see Proctor & Vu, 2006, for a review of compatibility principles). Compatibility effects are differences in speed and accuracy of responding as a function of how natural, or compatible, the relation between stimuli and responses is. Two types of compatibility effects can be distinguished (Kornblum, Hasbroucq, & Osman, 1990). For one type, certain sets of stimuli are more compatible with certain sets of responses than with others. For example, the combinations of verbal-vocal and spatial-manual sets yield better performance than the combinations of verbal-manual and spatial-vocal sets (Wang & Proctor, 1996). For the other type, within a specific stimulus-response set, some mappings of individual stimuli to responses produce better performance

than others. If one stimulus has the meaning "left" and the other "right," performance is better if the left stimulus is mapped to the left response and the right stimulus to the right response, for all stimulus and response modes.

Fitts and Seeger (1953) and Fitts and Deininger (1954) demonstrated both types of compatibility effects for spatially arranged display and response panels. However, compatibility effects occur for a much wider variety of other stimulus-response sets. According to Kornblum et al. (1990), dimensional overlap (similarity) between the stimulus and response sets is the critical factor. When the sets have dimensional overlap, a stimulus will activate its corresponding response automatically. If this response is correct (compatible mapping), responding will be facilitated, but if it is not correct (incompatible mapping), responding will be inhibited. A second factor contributing to the advantage for the compatible mapping is that intentional translation of the stimulus into a response will occur quicker when the mapping is compatible than when it is not. Most contemporary models of stimulus-response compatibility incorporate both automatic and intentional response-selection routes (Hommel & Prinz, 1997), although they differ regarding the exact conditions under which each plays a role and the way in which they interact.

One reason why automatic activation is considered to contribute to compatibility effects is that such effects occur when irrelevant stimulus information overlaps with the response set (Lu & Proctor, 1995). The Stroop color-naming effect, for which an incongruent color word produces interference in naming a relevant stimulus color, is the most well known example. An irrelevant stimulus location also produces interference when it is incongruent with the location of a key-press to a relevant stimulus dimension, a phenomenon known as the Simon effect (Simon, 1990). Psychophysiological studies in which the LRP has been measured have provided evidence that the Simon effect is due at least in part to activation of the response corresponding to stimulus location (Valle-Inclán, de Labra, & Redondo, 2000).

For completely unrelated stimulus and response sets that are structured, performance is better when structural correspondence is maintained (Reeve & Proctor, 1990). For instance, when stimuli and responses are ordered (e.g., a row of four stimulus locations and a row of four response locations), RT is faster when the stimulus-response mapping can be characterized by a rule (e.g., pressing the key at the mirror opposite location) than when the mapping is random (Duncan, 1977). Spatial compatibility effects also occur when display and response elements refer to orthogonal spatial dimensions (Cho & Proctor, 2003). However, stimulus-response compatibility effects sometimes do not occur under conditions in which one would expect them to. For example, when compatible and incompatible mappings are mixed within a single block, the typical compatibility effect is eliminated (Shaffer, 1965; Vu & Proctor, 2004). Moreover, the same display and response elements can be coded along multiple dimensions in certain situations (e.g., vertical position versus horizontal position). The relative importance of maintaining compatibility on each dimension is a function of how salient the dimensions are made by the task environment (Rubichi, Vu, Nicoletti, & Proctor, 2006).

Because when and where compatibility effects are going to occur is not always obvious, interface designers are likely to make poor decisions if they rely only on their intuitions. Payne (1995), Vu and Proctor (2003), and Tlauka (2004) have shown that naïve subjects can predict simple compatibility effects, such as that performance will be better with a mapping that is spatially compatible than with one that is not. However, they are not able to accurately predict many other compatibility effects that occur, such as the benefit of maintaining a consistent stimulus-response mapping rule. One encouraging finding is that estimates of relative compatibility can be improved by a small amount of experience performing with the different stimulus-response mappings (Vu & Proctor, 2003). The important point for HCI is that designers need to be aware of the potential problems created by various types of incompatibility between display and response elements because their effects are not always obvious. A designer can get a better feel for the relative compatibility of alternative arrangements by performing tasks that use them. However, after the designer selects a few arrangements that would seem to yield good performance, these remaining arrangements need to be tested more thoroughly on groups of users.

Response Execution

"Motor programming" refers to specification of the physical response that is to be made. This process is affected by variables such as relative stimulus-response frequency and movement direction. One factor that influences this stage is movement complexity. The longer the sequence of movements that is to be made upon occurrence of a stimulus in a choice-reaction task, the longer the RT to initiate the sequence (Sternberg, Monsell, Knoll, & Wright, 1978). This effect is thought to be due to the time required to load the movement sequence into a buffer before initiating the movements. Time to initiate the movement sequence decreases with practice, and recent fMRI evidence suggests that this decrease in RT involves distinct neural systems that support visuomotor learning of finger sequences and spatial learning of the locations of the finger movements on a keypad (Parsons, Harrington, & Rao, 2005).

One of the most widely known relations attributed to response execution is Fitts' Law, which describes the time to make aimed movements to a target location (Fitts, 1954). This law, as originally specified by Fitts, is:

$$\text{Movement Time} = a + b \log_2 (2D/W)$$

where a and b are constants, D is distance to the target, and W is target width. However, there are slightly different versions of the law. According to Fitts' Law, movement time is a direct function of distance and an inverse function of target width. Fitts' Law has been found to provide an accurate description of movement time in many situations, although alternatives have been proposed for certain situations. One of the factors that contribute to the increase in movement time as the index of difficulty increases is the need to make a corrective submovement based on feedback in order to hit the target location (Meyer, Abrams, Kornblum, Wright, & Smith, 1988).

The importance of Fitts' Law for HCI is illustrated by the fact that the December 2004 issue of *International Journal of Human-Computer Studies* was devoted to the 50th anniversary of

Fitts' original study. In the preface to the issue, the editors, Guiard and Beudouin-Lafon (2004), stated: "What has come to be known as Fitts' law has proven highly applicable in Human-Computer Interaction (HCI), making it possible to predict reliably the minimum time for a person in a pointing task to reach a specified target" (p. 747). Several illustrations of this point follow.

One implication of the law for interface design is that the slope of the function (*b*) may vary across different control devices, in which case movement times will be faster for the devices that yield lower slopes. Card, English, and Burr (1978) conducted a study that evaluated how efficient text keys, step keys, a mouse, and a joystick are at a text-selection task in which users selected text by positioning the cursor on the desired area and pressing a button or key. They showed that the mouse was the most efficient device for this task; positioning time for the mouse and joystick could be accounted for by Fitts' Law, with the slope of the function being less steep for the mouse; positioning time with the keys was proportional to the number of keystrokes that had to be executed.

Another implication of Fitts' Law is that any reduction in the index of difficulty should decrease the time for movements. Walker, Smelcer, and Nilsen (1991) evaluated movement time and accuracy of menu selection for the mouse. Their results showed that reducing the distance to be traveled (which reduces the index of difficulty) by placing the initial cursor in the middle of the menu, rather than the top, improved movement time. Placing a border around the menu item in which a click would still activate that item, and increasing the width of the border as the travel distance increases, also improved performance. The reduction in movement time by use of borders is predicted by Fitts' Law because borders increase the size of the target area.

Gillan, Holden, Adam, Rudisill, and Magee (1992) noted that designers must be cautious when applying Fitts' Law to HCI because factors other than distance and target size play a role when using a mouse. Specifically, they proposed that the critical factors in pointing and dragging are different than those in pointing and clicking (which was the main task in Card et al.'s (1978) study). Gillan et al. showed that, for a text-selection task, both point-click and point-drag movement times can be accounted for by Fitts' Law. For point-click sequences, the diagonal distance across the text object, rather than the horizontal distance, provided the best fit for pointing time. For point-drag, the vertical distance of the text provided the best fit. The reason why the horizontal distance is irrelevant is that the cursor must be positioned at the beginning of the string for the point-drag sequence. Thus, task requirements should be taken into account before applying Fitts' Law to the interface design.

Motor adjustment deals with the transition from a central motor program to peripheral motor activity. Studies of motor adjustment have focused on the influence of foreperiod duration on motor preparation. In a typical study, a neutral warning signal is presented at various intervals prior to the onset of the imperative stimulus. Bertelson (1967) varied the duration of the warning foreperiod and found that RT reached a minimum at a foreperiod of 150 ms and then increased slightly at the 200- and 300-ms foreperiods. However, error rate increased to a maximum at the 150-ms foreperiod and decreased slightly at the longer foreperiods. This relatively typical pattern suggests that it takes time to attain a state of high motor preparation, and that this state reflects an increased readiness to respond quickly at the expense of accuracy. Courtire, Hardouin, Vidal, Possamai, and Hasbroucq (2003) recently concluded that nitrous oxide impaired motor adjustment in rats because inhalation of nitrous oxide interacted with foreperiod duration but not with stimulus luminance.

MEMORY IN INFORMATION PROCESSING

Memory refers to explicit recollection of information in the absence of the original stimulus and to persisting effects of that information on information processing that may be implicit. Memory may involve recall of an immediately preceding event or one many years in the past, knowledge derived from everyday life experiences and education, or procedures learned to accomplish complex perceptual-motor tasks. Memory can be classified into several categories. Episodic memory refers to memory for a specific event such as going to the movie last night, whereas semantic memory refers to general knowledge such as what a movie is. Declarative memory is verbalizable knowledge, and procedural memory is knowledge that can be expressed nonverbally. In other words, declarative memory is knowledge that something is the case, whereas procedural memory is knowledge of how to do something. For example, telling your friend your new phone number involves declarative memory, whereas riding a bicycle involves procedural knowledge. A memory test is regarded as explicit if a person is asked to judge whether a specific item or event has occurred before in a particular context; the test is implicit if the person is to make a judgment, such as whether a string of letters is a word or nonword, that can be made without reference to earlier "priming" events. In this section, we focus primarily on explicit episodic memory.

Three types of memory systems are customarily distinguished: sensory stores, short-term memory (STM; or working memory), and long-term memory (LTM). Sensory stores, which we will not cover in detail, refer to brief modality-specific persistence of a sensory stimulus from which information can be retrieved for one or two seconds (see Nairne, 2003). STM and LTM are the main categories by which investigations of episodic memory are classified, and as the terms imply, the distinction is based primarily on duration. The dominant view is that these are distinct systems that operate according to different principles, but there has been debate over whether the processes involved in these two types of memories are the same or different. A recent fMRI study by Talmi, Grady, Goshen-Gottstein, and Moscovitch (2005) found that recognition of early items in the list was accompanied by activation of areas in the brain associated with LTM whereas recognition of recent items did not, supporting a distinction between STM and LTM stores.

Short-Term (Working) Memory

STM refers to representations that are currently being used or have recently been used and last for a short duration. A distinguishing characteristic is that STM is of limited capacity. This point was emphasized in G. A. Miller's (1956) classic article,

"The Magical Number Seven Plus or Minus Two," in which he indicated that capacity is not simply a function of the number of items, but rather the number of "chunks." For example, "i," "b," and "m" are three letters, but most people can combine them to form one meaningful chunk: "IBM." Consequently, memory span is similar for strings of unrelated letters and strings of meaningful acronyms or words. Researchers refer to the number of items that can be recalled correctly, in order, as "memory span."

As most people are aware from personal experience, if distracted by another activity, information in STM can be forgotten quickly. With respect to HCI, Oulasvirta and Saariluoma (2004) noted that diversion of attention from the current task to a competing task is a common occurrence, for example, when an unrequested pop-up dialogue box requiring an action appears on a computer screen. Laboratory experiments have shown that recall of a string of letters that is within the memory span decreases to close to chance levels over a retention interval of 18 s when rehearsal is prevented by an unrelated distractor task (Brown, 1958; Peterson & Peterson, 1959). This short-term forgetting was thought initially to be a consequence of decay of the memory trace due to prevention of rehearsal. However, Keppel and Underwood (1962) showed that proactive interference from items on previous lists is a significant contributor to forgetting. They found no forgetting at long retention intervals when only the first list in a series was examined, with the amount of forgetting being much larger for the second and third lists as proactive interference built up. Consistent with this interpretation, "release" from proactive inhibition—that is, improved recall—occurs when the category of the to-be-remembered items on the current list differs from that of previous lists (D. D. Wickens, 1970).

The capacity limitation of STM noted by G. A. Miller (1956) is closely related to the need to rehearse the items. Research has shown that the memory span, the number of words that can be recalled correctly in order, varies as a function of word length. That is, the number of items that can be retained decreases as word length increases. Evidence has indicated that the capacity is the number of syllables that can be said in about two seconds (Baddeley, Thomson, & Buchanan, 1975; Schweickert & Boruff, 1986). That pronunciation rate is critical suggests a time-based property of STM, which is consistent with a decay account. Consequently, the most widely accepted view is that both interference and decay contribute to short-term forgetting, with decay acting over the first few seconds and interference accounting for the largest part of the forgetting.

As the complexity of an HCI task increases, one consequence is the overloading of STM. Jacko and Ward (1996) varied four different determinants of task complexity (multiple paths, multiple outcomes, conflicting interdependence among paths, or uncertain or probabilistic linkages) in a task requiring use of a hierarchical menu to acquire specified information. When one determinant was present, performance was slowed by approximately 50%, and when two determinants were present in combination, performance was slowed further. That is, as the number of complexity determinants in the interface increased, performance decreased. Jacko and Ward attributed the decrease in performance for all four determinants to the increased STM load they imposed.

The best-known model of STM is Baddeley and Hitch's (1974) working memory model, which partitions STM into three main parts: central executive, phonological loop, and visuospatial sketchpad. The central executive controls and coordinates the actions of the phonological loop and visuospatial sketchpad. The phonological loop is composed of a phonological store that is responsible for storage of the to-be-remembered items, and an articulatory control process that is responsible for recoding verbal items into a phonological form and rehearsal of those items. The items stored in the phonological store decay over a short interval and can be refreshed through rehearsal from the articulatory control process. The visuospatial sketchpad retains information regarding visual and spatial information, and it is involved in mental imagery.

The working memory model has been successful in explaining several phenomena of STM (Baddeley, 2003)—for example, that the number of words that can be recalled is affected by word length. However, the model cannot explain why memory span for visually presented material is only slightly reduced when subjects engage in concurrent articulatory suppression (such as saying the word "the" aloud repeatedly). Articulatory suppression should monopolize the phonological loop, preventing any visual items from entering it. To account for such findings, Baddeley revised the working memory model to include an episodic buffer (see Fig. 2.3). The buffer is a limited-capacity temporary store that can integrate information from the phonological loop, visuospatial sketchpad, and long-term memory. By attending to a given source of information in the episodic buffer, the central executive can create new cognitive representations that might be useful in problem solving.

Long-Term Memory

LTM refers to representations that can be remembered for durations longer than can be attributed to STM. LTM can involve information presented from minutes to years ago. Initially, it was

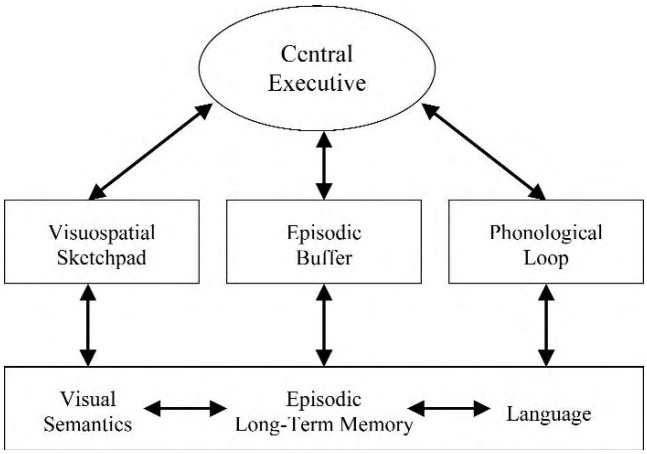


FIGURE 2.3. Baddeley's (2000) revised working memory model. From "The episodic buffer: A new component of working memory?" by A. D. Baddeley, 2000, *Trends in Cognitive Sciences*, 4, p. 421. Copyright 2000 by Elsevier Science Ltd. with permission.

thought that the probability of an item being encoded into LTM was a direct function of the amount of time that it was in STM, or how much it was rehearsed. However, Craik and Watkins (1973) showed that rehearsal in itself is not sufficient, but rather that deep-level processing of the meaning of the material is the important factor in transferring items to LTM. They presented subjects with a list of words and instructed them that when the experimenter stopped the presentation, they were to recall the last word starting with the letter "a." The number of other words between instances of "a" words was varied, with the idea that the amount of time a word was rehearsed would depend on the number of words before the next "a" word. At the end of the session, subjects were given a surprise test in which they were to recall all "a" words. There was no effect of number of intervening words on recall, suggesting that although subjects rehearsed the words longer, their recall did not improve because the words were not processed deeply.

Craik and Watkins' (1973) results were consistent with the levels of processing framework proposed by Craik and Lockhart (1972). According to this view, encoding proceeds in a series of analyses, from shallow perceptual features to deeper, semantic levels. The deeper the level of processing, the more strongly the item is encoded in memory. A key study supporting the levels of processing view is that of Hyde and Jenkins (1973). In their study, groups of subjects were presented a list of words for which they engaged in shallow processing (e.g., deciding whether each word contained a capital letter) or deep processing (e.g., identifying whether each word was a verb or a noun). Subjects were not told in advance that they would be asked to recall the words, but were given a surprise recall test at the end of the session. The results showed that the deep processing group recalled more words than the shallow processing group. Of direct relevance to HCI, Oulasvirta, Kärkkäinen, and Laarni (2005) recently found that participants who viewed the content area of a web page had no better memory for the locations of content objects than did a control group who guessed where those objects would be placed, because the participants' task was to locate navigational objects on the page and not to process the content information.

Another well-known principle for LTM is encoding specificity, which states that the probability that a retrieval cue results in recollection of an earlier event is an increasing function of the match between the features encoded initially and those provided by the retrieval cue (Tulving & Thomson, 1973). An implication of this principle is that memory will be context dependent. Godden and Baddeley (1975) demonstrated a context-dependent memory effect by having divers learn a list of words on land or under water, and recall the words on land or under water. Recall was higher for the group who learned on land when the test took place on land than under water, and vice versa for the group who learned under water. A related principle is that of transfer-appropriate processing (Morris, Bransford, & Franks, 1977). Morris et al. showed that deep-level semantic judgments during study produced better performance than shallow rhyme judgments on a standard recognition memory test. However, when the memory test required decisions about whether the test words rhymed with studied words, the rhyme judgments led to better performance than the semantic judgments. Healy, Wohldmann, and Bourne (2005) have proposed that encoding specificity and transfer-appropriate processing can be incorporated within the single principle of procedural reinstatement: Retention will be evident to the extent that the procedures engaged in during study or training are reinstated during the retention test.

Research has confirmed that the levels-of-processing framework must accommodate the effects of the retention context, as captured by the above principles, to explain the effects of processing performed during encoding. Although levels-of-processing has a strong effect on accuracy of explicit recall and recognition, Jacoby and Dallas (1981) found no effect on an implicit memory test. Later studies have shown a robust effect of levels-of-processing on implicit tests similar to that obtained for recall and recognition if the test is based on conceptual cues, rather than on perceptual cues (see Challis, Velichkovsky, & Craik, 1996). Challis et al. constructed direct recognition tests, in which the words were graphemically, phonologically, or semantically similar to the studied words, that showed no levels of processing effect. They emphasized that, to account for levels-of-processing results, it is necessary to specify the types of information produced by the levels-of-processing, the types of information required for the specific test, and how task instructions modify encoding and retrieval processes.

Other Factors Affecting Retrieval of Earlier Events

Memory researchers have studied many factors that influence long-term retention. Not surprisingly, episodic memory improves with repetition of items or events. Also, massed repetition (repeating the same item in a row) is less effective than spaced repetition (repeating the same item with one or more intervening items). This benefit for spaced repetition, called the "spacing effect" or "lag effect," is often attributed to two main factors. First, study time for the same items appearing in succession is less than study time for the same items appearing further apart. Second, when the items are studied over a longer period of time, there is an opportunity for the items to be associated with different cues that can aid recall later. The spacing or lag effect is widespread and occurs for both recall and recognition (Hintzman, 1974). Bahrick and Hall (2005) noted that a similar spacing benefit is found for learning lists of items when practice sessions, each with test and learning phases, are separated by several days. They presented evidence that a large part of the spacing benefit in this case arises from individuals determining which study strategies are more effective at promoting long-term retention and then using those strategies more.

Another widely studied phenomenon is the generation effect, in which recall is better when subjects have to generate the to-be-remembered words rather than just studying the words as they are presented (Slamecka & Graf, 1978). In a generation effect experiment, subjects are divided into two groups, read and generate. Each group receives a series of words, with each word spelled out completely for the read group and missing letters for the generate group. An example is as follows:

Read group:	CAT; ELEPHANT; GRAPE; CAKE
Generate group:	C _ T; E_E_H _ _ NT; G _ APE; CAK_

The typical results show that subjects in the generate group can recall more words than those in the read group. One application of the generation effect to HCI is that when a computer user needs a password for an account, the system should allow the user to generate the password rather than providing her with one because the user would be more likely to recall the generated password. The common method of proactive password generation, in which users are asked to generate a password that meets certain restrictions (e.g., contain an uppercase letter, a lowercase letter, a digit, etc.), is intended to result in more memorable and secure passwords (e.g., Vu, Bhargav, & Proctor, 2003).

Events that precede or follow an event of interest can interfere with recall of that event. The former is referred to as "proactive interference," and was discussed in the section on STM; the latter is referred to as "retroactive interference." One area of research in which retroactive interference is of central concern is that of eyewitness testimony. Loftus and Palmer (1974) showed that subsequent events could distort a person's memory of an event that the person witnessed. Subjects were shown a sequence of events depicting a car accident. Subsequently, they were asked the question, "How fast were the cars going when they _____ each other." When the verb "contacted" was used, subjects estimated the speed to be 32 miles per hour, and only one-tenth of them reported seeing broken glass. However, when the verb "smashed" was used, the estimated speed increased to 41 miles per hour, and almost one third of the subjects reported seeing broken glass. Demonstrations like these indicate not only that retroactive interference can cause forgetting of events, but that it also can cause the memory of events to be changed. More recent research has shown that completely false memories can be implanted (see Roediger & McDermott, 1995).

Mnemonic techniques can also be used to improve recall. The basic idea behind mnemonics is to connect the to-be-remembered material with an established organizational structure that can be easily accessible later on. Two widely used mnemonic techniques are the pegword method (Wood & Pratt, 1987) and the method of loci (Verhaeghen & Marcoen, 1996). In the pegword method, a familiar rhyme provides the organizational structure. A visual image is formed between each pegword in the rhyme and the associated target item. At recall, the rhyme is generated, and the associated items come to mind. For the method of loci, locations from a well-known place, such as your house, are associated with the to-be-remembered items. Although specific mnemonic techniques are limited in their usefulness, the basic ideas behind them (utilizing imagery, forming meaningful associations, and using consistent encoding and retrieval strategies) are of broad value for improving memory performance.

Vu, Tai, Bhargav, Schultz, and Proctor (2004) examined the effectiveness of a "first-letter" mnemonic technique to help users relate individual characters of a password to a structured sentence in order to aid recall at a later time. In one condition, Vu et al. had users generate a sentence and take the first letter of each word in the sentence to form a password; in another condition, users generated a sentence that also included numbers and special characters embedded into the sentence and resulting password. Passwords generated using the first-letter technique were more memorable when users did not have to embed a digit and special character into the sentence, but were more secure (i.e., more resistant to cracking) when the sentence and resulting password included the digit and special character. Thus, when it comes to memory and security of computer passwords, there seems to be a tradeoff between memorability and security.

ATTENTION IN INFORMATION PROCESSING

Attention is increased awareness directed at a particular event or action to select it for increased processing. This processing may result in enhanced understanding of the event, improved performance of an action, or better memory for the event. Attention allows us to filter out unnecessary information so that we can focus on a particular aspect that is relevant to our goals. Several significant information-processing models of attention have been proposed.

Models of Attention

In an influential study, Cherry (1953) presented subjects with different messages to each ear through headphones. Subjects were to repeat aloud one of the two messages while ignoring the other. When subsequently asked questions about the two messages, subjects were able to accurately describe the message to which they were attending but could not describe anything except physical characteristics, such as gender of the speaker, about the unattended message.

To account for such findings, Broadbent (1958) developed the filter theory, which assumes that the nervous system acts as a single-channel processor. According to filter theory, information is received in a preattentive temporary store and then is selectively filtered, based on physical features such as spatial location, to allow only one input to access the channel. Broadbent's filter theory implies that the meaning of unattended messages is not identified, but later studies showed that the unattended message could be processed beyond the physical level, in at least some cases (Treisman, 1964).

To accommodate the finding that meaning of an unattended message can influence performance, Treisman (1964) reformulated filter theory into what is called the "filter-attenuation theory." According to attenuation theory, early selection by filtering still precedes stimulus identification, but the filter only attenuates the information on unattended channels. This attenuated signal may be sufficient to allow identification if the stimulus is one with a low identification threshold, such as a person's name or an expected event. Deutsch and Deutsch (1963) proposed that unattended stimuli are always identified and the bottleneck occurs in later processing, a view called "late-selection theory." The difference between attenuation theory and late-selection theory and is that latter assumes that meaning is fully analyzed, but the former does not.

Lavie, Hirst, de Fockert, and Viding (2004) have proposed a load theory of attention, which they claim "resolves the long-standing early versus late selection debate" (p. 339). Specifically,

the load theory includes two selective attention mechanisms: a perceptual selection mechanism and a cognitive control mechanism. When perceptual load is high (i.e., great demands are placed on the perceptual system), the perceptual mechanism excludes irrelevant stimuli from being processed. When memory load is high, it is not possible to suppress irrelevant information at a cognitive level. In support of load theory, Lavie et al. showed that interference from distracting stimuli is reduced under conditions of high perceptual load but increased under conditions of high working memory load.

In divided attention tasks, a person must attend to multiple sources of information simultaneously. Kahneman (1973) proposed a unitary resource model that views attention as a single resource that can be divided up among different tasks in different amounts, based on task demands and voluntary allocation strategies. Unitary resource models provided the impetus for dual-task methodologies (such as performance operating characteristics) and mental workload analyses that are used widely in HCI (Eberts, 1994). The expectation is that multiple tasks should produce interference when their resource demands exceed the supply that is available.

Many studies have shown that it is easier to perform two tasks together when they use different stimulus or response modalities than when they use the same modalities. Performance is also better when one task is verbal and the other visuospatial than when they are the same type. These result patterns provide the basis for multiple resource models of attention such as that of C. D. Wickens (1984). According to multiple resource models, different attentional resources exist for different sensory-motor modalities and coding domains. Multiple resource theory captures the fact that multiple task performance typically is better when the tasks use different input-output modes than when they use the same modes. However, it is often criticized as being too flexible because new resources can be proposed arbitrarily to fit any finding of specificity of interference (Navon, 1984).

A widely used metaphor for visual attention is that of a spotlight that is presumed to direct attention to everything in its field (Posner & Cohen, 1984). Direction of attention is not necessarily the same as the direction of gaze because the attentional spotlight can be directed independently of fixation. Studies show that when a location is cued as likely to contain a target stimulus, but then a probe stimulus is presented at another location, a spatial gradient surrounds the attended location such that items nearer to the focus of attention are processed more efficiently than those farther away from it (Yantis, 2000). The movement of the attentional spotlight to a location can be triggered by two types of cues, exogenous and endogenous. An exogenous cue is an external event such as the abrupt onset of a stimulus at a peripheral location that involuntarily draws the attentional spotlight to its location. Exogenous cues produce rapid performance benefits, which dissipate quickly, for stimuli presented at the cued location. This is followed by a period in which performance is worse for stimuli at the cued location than for ones presented at the uncued location, a phenomenon called "inhibition of return" (Posner & Cohen, 1984). An endogenous cue is typically a symbol such as a central arrowhead that must be identified before a voluntary shift in attention to the designated location can be made. The performance benefits for endogenous cues take longer to develop and are sustained for a longer period of time when the cues are relevant, indicating that their benefits are due to conscious control of the attentional spotlight (Klein & Shore, 2000).

In a visual search task, subjects are to detect whether a target is present among distractors. Treisman and Gelade (1980) developed Feature Integration Theory to explain the results from visual search studies. When the target is distinguished from the distractors by a basic feature such as color (feature search), RT and error rate often show little increase as the number of distractors increases. However, when two or more features must be combined to distinguish the target from distractors (conjunctive search), RT and error rate typically increase sharply as the number of distractors increases. To account for these results, feature integration theory assumes that basic features of stimuli are encoded into feature maps in parallel across the visual field at a preattentive stage. Feature search can be based on this preattentive stage because a "target-present" response requires only detection of the feature. The second stage involves focusing attention on a specific location and combining features that occupy the location into objects. Attention is required for conjunctive search because responses cannot be based on detection of a single feature. According to feature integration theory, performance in conjunctive search tasks decreases as the number of distractors increases because attention must be moved sequentially across the search field until a target is detected or all items present have been searched. Feature integration theory served to generate a large amount of research on visual search that showed, as is typically the case, that the situation is not as simple as depicted by the theory. This has resulted in modifications of the theory, as well as alternative theories. For example, Wolfe's (1994) Guided Search Theory maintains the distinction between an initial stage of feature maps and a second stage of attentional binding, but assumes that the second stage is guided by the initial feature analysis.

In HCI, a common visual search task involves locating menu items. When users know exactly what option to search for, they can use identity matching, in which they search the display for the menu name that they want to find. Perlman (1984) suggested that when identity search is used, the menu options should be displayed in alphabetical order to facilitate search. When users do not know where an option is included within a main list of menus, inclusion matching is used. The users must decide in which group the specific option would be categorized and then search the list of items within that group. With inclusion matching, search times may be longer for items that can be classified in more than one of the main groupings or when the items are less well-known examples of a main grouping (Somberg & Picardi, 1983). Equivalence search occurs when the user knows what option to select, but does not know how that option is labeled. McDonald, Stone, and Liebelt (1983) showed that alphabetical and categorical organizations yield shorter search times than randomized organization for equivalence search. Search can also be affected by the breadth versus depth of the menu design. Lee and MacGregor (1985) showed that deep hierarchies are preferred over broad ones. However, more recently, Tullis, Catani, Chadwick-Dias, and Cianchette (2005) suggested that, for complex or ambiguous situations, there is a benefit for broad menu designs because they facilitate comparison between categories. The main point is that, when struc-

turing menus, designers must take into account the type of search in which the user would most likely be engaged.

The role of attention in response selection has been investigated extensively using the psychological refractory period (PRP) paradigm (Pashler, 1998). In the PRP paradigm, a pair of choice-reaction tasks must be performed, and the stimulus onset asynchrony (SOA) of the second stimulus is presented at different intervals. RT for Task 2 is slowed at short SOAs, and this phenomenon is called the "PRP effect." The experimental results have been interpreted with what is called "locus of slack logic" (Schweickert, 1978), which is an extension of additive factors logic to dual-task performance. The basic idea is that if a Task 2 variable has its effect prior to a bottleneck, that variable will have an underadditive interaction with SOA. This underadditivity occurs because, at short SOAs, the slack period during which post-bottleneck processing cannot begin can be used for continued processing for the more difficult condition. If a Task 2 variable has its effect after the bottleneck, the effect will be additive with SOA.

The most widely accepted account of the PRP effect is the response-selection bottleneck model (Pashler, 1998). The primary evidence for this model is that perceptual variables typically have underadditive interactions with SOA, implying that their effects are prior to the bottleneck. In contrast, post-perceptual variables typically have additive effects with SOA, implying that their effects are after the bottleneck. There has been dispute as to whether there is also a bottleneck at the later stage of response initiation (De Jong, 1993); whether the response-selection bottleneck is better characterized as a parallel processor of limited capacity that divides resources among to-be-performed tasks (Tombu & Jolicœur, 2005); and whether the apparent response-selection bottleneck is structural, or simply a strategy adopted by subjects to comply with task instructions (Meyer & Kieras, 1997). This latter approach is consistent with a recent emphasis on the executive functions of attention in the coordination and control of cognitive processes (Monsell & Driver, 2000).

Automaticity and Practice

Attention demands are high when a person first performs a new task. However, these demands decrease and performance improves as the task is practiced. Because the quality of performance and attentional requirements change substantially as a function of practice, it is customary to describe performance as progressing from an initial cognitively demanding phase to a phase in which processing is automatic (Anderson, 1982; Fitts & Posner, 1967).

The time to perform virtually any task from choice RT to solving geometry problems decreases with practice, with the largest benefits occurring early in practice. Newell and Rosenbloom (1981) proposed a power function to describe the changes in RT with practice:

$$RT = BN^{-\alpha}$$

where N is the number of practice trials, B is RT on the first trial, and α is the learning rate. Although the power function has become widely accepted as a law that describes the changes in RT, Heathcote, Brown, and Mewhort (2000) indicated that it does not fit the functions for individual performers adequately. They showed that exponential functions provided better fits than power functions to 40 individual datasets, and proposed a new exponential law of practice. The defining characteristic of the exponential function is that the relative learning rate is a constant at all levels of practice, whereas, for the power function, the relative learning rate is a hyperbolically decreasing function of practice trials.

PROBLEM SOLVING AND DECISION MAKING

Beginning with the work of Newell and Simon (1972), it has been customary to analyze problem solving in terms of a problem space. The problem space consists of the following: (a) an initial state, (b) a goal state that is to be achieved, (c) operators for transforming the problem from the initial state to the goal state in a sequence of steps, and (d) constraints on application of the operators that must be satisfied. The problem-solving process itself is conceived of as a search for a path that connects the initial and goal states.

Because the size of a problem space increases exponentially with the complexity of the problem, most problem spaces are well beyond the capacity of short-term memory. Consequently, for problem solving to be effective, search must be constrained to a limited number of possible solutions. A common way to constrain search is to use heuristics. For example, people often use a means-ends heuristic for which, at each step, an operator is chosen that will move the current state closer to the goal state (Atwood & Polson, 1976). Such heuristics are called "weak methods" because they do not require much knowledge about the exact problem domain. Strong methods, such as those used by experts, rely on prior domain-specific knowledge and do not require much search because they are based on established principles applicable only to certain tasks.

The problem space must be an appropriate representation of the problem if the problem is to be solved. One important method for obtaining an appropriate problem space is to use analogy or metaphor. Analogy enables a shift from a problem space that is inadequate to one that may allow the goal state to be reached. There are several steps in using analogies (Holland, Holyoak, Nisbett, & Thagard, 1986), including detecting similarity between source and target problems, and mapping the corresponding elements of the problems. Humans are good at mapping the problems, but poor at detecting that one problem is an analog of another. An implication for HCI is that potential analogs should be provided to users for situations in which they are confronted by novel problems.

The concept of mental model, which is closely related to that of the problem space, has become widely used in recent years (see chapter 3, this volume). The general idea of mental models with respect to HCI is that as the user interacts with the computer, he or she receives feedback from the system that allows him or her to develop a representation of how the system is functioning for a given task. The mental model incorporates the goals of the user, the actions taken to complete the goals, and

expectations of the system's output in response to the actions. A designer can increase the usability of an interface by using metaphors that allow transfer of an appropriate mental model (e.g., the desktop metaphor), designing the interface to be consistent with other interfaces with which the user is familiar (e.g., the standard web interface), and conveying the system's functions to the user in a clear and accurate manner. Feedback to the user is perhaps the most effective way to communicate information to the user and can be used to guide the user's mental model about the system.

Humans often have to make choices regarding situations in which the outcome depends on events that are outside of those humans' control. According to expected utility theory, a normative theory of decision making under uncertainty, the decision maker should determine the expected utility of a choice by multiplying the subjective utility of each outcome by the outcome's probability and summing the resulting values (see Proctor & Van Zandt, 1994). The expected utility should be computed for each choice, and the optimal decision is the choice with the highest expected utility. It should be clear from this description that for all but the simplest of problems, a human decision maker cannot operate in this manner. To do so would require attending to multiple cues that exceed attentional capacity, accurate estimates of probabilities of various events, and maintenance of, and operation on, large amounts of information that exceed short-term memory capacity.

Research of Kahneman and Tversky (2000) and others has shown that what people do when the outcome associated with a choice is uncertain is to rely heavily on decision-making heuristics. These heuristics include representativeness, availability, and anchoring. The representativeness heuristic is that the probability of an instance being a member of a particular category is judged on the basis of how representative the instance is of the category. The major limitation of the representativeness heuristic is that it ignores base rate probabilities for the respective categories. The availability heuristic involves determining the probability of an event based on the ease with which instances of the event can be retrieved. The limitation is that availability is affected not only by relative frequency, but also by other factors. The anchoring heuristic involves making a judgment regarding probabilities of alternative states based on initial information, and then adjusting these probabilities from this initial "anchor" as additional information is received. The limitation of anchoring is that the initial judgment can produce a bias for the probabilities. Although heuristics are useful, they may not always lead to the most favorable decision. Consequently, designers need to make sure that the choice desired for the user in a particular situation is one that is consistent with the user's heuristic biases.

NEW DEVELOPMENTS AND ALTERNATIVE APPROACHES

It should not be surprising that since its ascendance, the human information-processing approach has been challenged on many grounds, and alternative approaches have been championed. We describe three of those approaches below. In advocating one, Smith and Henning (2005) recently stated, "The basic scientific problem with information processing models that have come to dominate cognitive science in the past four decades is that they are non-refutable" (p. 641). Such an argument misses the point that although the theoretical framework itself is non-refutable, as with any other such framework, the specific models and theories developed within the framework are refutable. The issue is the relative success of the information-processing approach at providing solutions to issues of theoretical and applied importance. Pashler (1998), who took an information-processing approach to the study of attention, noted, "The success or failure of information-processing psychology can be assessed only on the basis of insights that do or do not emerge from the research it spawns" (p. 7). It is the assessment of many researchers that the information-processing approach has generally fared well on many counts in comparison to alternative approaches.

Cognitive Neuroscience

A quick scan of journals and job announcements would reveal that cognitive neuroscience is a rapidly emerging field. As noted earlier, the major goal of research in this area is to analyze the brain mechanisms responsible for cognitive functioning. The knowledge gained from behavioral studies of human information processing provides the foundation for much of this research. This point is acknowledged explicitly in the instructions for authors for the *Journal of Cognitive Neuroscience*, which state, "The Journal publishes papers that bridge the gap between descriptions of information processing and specifications of brain activity" (MIT Press, 2007). Note that there would not be a gap to bridge were it not for the success of research on human information processing. If this gap is bridged successfully, the advances in neuroergonomics and augmented cognition envisioned by Parasuraman (2003) and Schmorrow (2005) will become reality.

The Ecological Approach

Gibson (1979) developed an ecological approach to perception and action that is typically presented as an alternative to the information-processing approach (e.g., Schvaneveldt, 2005). The ecological approach places emphasis on analyzing the information that is available in the optic array and the dynamics of this information as individuals and objects move in the environment. It also takes the position that perception is direct and that accounts relying on mental representations and cognitive processes are unnecessary and incorrect (see, for example, Michaels & Stins, 1997). Much of HCI involves performing information-intensive tasks in artificial, human-made environments that do not resemble carrying out actions in the natural environment. For example, Aziz and Macredie (2005) wrote, "The use of web-based information system[s] mainly involves processing of information" (p. 1). Because such environments are similar to those studied in basic human information-processing experiments, the approach has much to offer in analyzing and understanding the tasks performed in them.

For HCI tasks of a more ecological nature, such as navigating in a virtual world, understanding the information available to an observer/actor in the environment being simulated is clearly essential. As an example, Schvaneveldt (2005) noted that he and his colleagues have focused their recent studies of aviation on analyses of the information in the flight environment that allows pilots to perform effectively during different stages of flight. From an information-processing perspective, although such ecological analyses are valuable, and indeed necessary, they must be incorporated within information-processing models (e.g., Ullman, 1980).

Cybernetic Approach

The cybernetic view of cognition is that cognition emerges as a consequence of motor control over sensory feedback. The cybernetic approach is closely related to the ecological approach, but places greater emphasis on self-regulated control of perception and cognition. Smith and Henning (2005), who advocated a cybernetic approach, took a strong position against the information-processing approach, stated:

> The cybernetic perspective presented here differs in a number of fundamental scientific respects from an information processing model. Behavioral cybernetics . . . emphasizes active control of information as sensory feedback via motor-sensory behavior, with motor-sensory behavior mediating both perception and cognition. In contrast, information processing models treat information as a fixed commodity presented on the input side of the processing system, ignoring the need for specific design factors to promote human motor control over this information as a source of sensory feedback. This failure to include motor-sensory control is a direct consequence of using an information processing model, where overt/motor behavior is viewed as a system output with no central role in the organization and control over information as feedback, nor over the control of subsequent behavior through the reciprocal effects of motor control on psychophysiological state. (p. 641)

Smith and Henning's (2005) statement stands in contrast to Young et al.'s (2004) assessment, described earlier in the chapter, that an information-processing analysis is essential to understanding the dynamic interactions of an operator with a vehicle for purposes of computer-aided augmented cognition. Although Smith and Henning may be correct that greater emphasis should be placed on the role of sensory feedback produced by actions, they have mistakenly attributed a simplifying tactic used by researchers to fundamental assumptions underlying the human information-processing approach. Recollect that human information processing has its roots in control theory.

Situated Cognition

Another approach that has been advocated recently is that of situated cognition, or situated action. Kiekel and Cooke (2005) stated that, according to situated cognition theories, "[m]uch of what symbolic [information-processing] theorists assign to the individual's head takes place outside of the confines of the individual and is directed by a world that is a much richer place than symbolic theories tend to suggest" (p. 92). They go on to say, "Information processing research tends to isolate psychological principles from a generic context, such as a laboratory . . ., while SA [situated action] research focuses on understanding the specific contextual constraints of the environment" (pp. 92–93). The extent to which the principles and theories developed from information-processing research conducted in the laboratory generalize to other contexts is an empirical issue. Although the evidence is not all in, the widespread application of information-processing principles and theories to human factors and HCI (e.g., Wickens, Lee, Liu, & Becker, 2004) suggests that what has been learned about human information processing is applicable to a variety of domains.

SUMMARY AND CONCLUSION

The methods, theories, and models in human information processing are currently well developed. The knowledge in this area, which we are only able to describe at a surface level in this chapter, is relevant to a wide range of concerns in HCI, from visual display design to representation and communication of knowledge. For HCI to be effective, the interaction must be made compatible with the human information-processing capabilities. Cognitive architectures that incorporate many of the facts about human information processing have been developed that can be applied to HCI. The Model Human Processor of Card et al. (1983) is the most widely known, but applications of other more recent architectures, including the ACT model of Anderson and colleagues (Anderson, Matessa, & Lebiere, 1997), the SOAR Model of Newell and colleagues (Howes & Young, 1997), and the EPIC Model of Kieras and Meyer (1997), hold considerable promise for the field, as demonstrated in chapter 5 of this volume.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*, 369–406.

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction, 12*, 439–462.

Atwood, M. E., & Polson, P. G. (1976). A process model for water jug problems. *Cognitive Psychology, 8*, 191–216.

Aziz, M., & Macredie, R. (2005). Specifying a model to explore the role of perceived usefulness in web-based information System use. In G. Salvendy (Ed.), *HCI International 2005.* Mahwah, NJ: Erlbaum.

Baddeley, A. (2003). Working memory and language: An overview. *Journal of Communication Disorders, 36*, 189–208.

Baddeley, A. D., & Hitch, G. J. (1974). Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 8, pp. 47–89). New York: Academic Press.

Baddeley, A. D., Thomson, N., & Buchanan, M. (1975). Word length and the structure of shortterm memory. *Journal of Verbal Learning and Behavior, 14*, 575–589.

Bahrick, H. P., & Hall, L. K. (2005). The importance of retrieval failures to long-term retention: A metacognitive explanation of the spacing effect. *Journal of Memory and Language, 52*, 566–577.

Balakrishnan, J. D. (1998). Measures and interpretations of vigilance performance: Evidence against the detection criterion. *Human Factors, 40*, 601–623.

Bertelson, P. (1967). The time course of preparation. *Quarterly Journal of Experimental Psychology, 19*, 272–279.

Boldini, A., Russo, R., & Avons, S. E. (2004). One process is not enough! A speed-accuracy tradeoff study of recognition memory. *Psychonomic Bulletin & Review, 11*, 353–361.

Bonin-Guillaume, S., Possamäi, C.-A., Blin, O., & Hasbroucq, T. (2000). Stimulus preprocessing, response selection, and motor adjustment in the elderly: An additive factor analysis. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition, 19*, 245–255.

Bower, G. H., & Hilgard, E. R. (1981). *Theories of learning* (5th ed.). Englewood Cliffs, NJ: Prentice-Hall.

Broadbent, D. E. (1958). *Perception and communication*. Oxford, UK: Pergamon Press.

Brown, J. (1958). Some tests of the decay theory of immediate memory. *Quarterly Journal of Experimental Psychology, 10*, 12–21.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of the mouse, rate-controlled isometrick joystick, step keys, and text keys for text selection on a CRT. *Ergonomics, 21*, 601–613.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Challis, B. H., Velichkovsky, B. M., & Craik, F. I. M. (1996). Levels-of-processing effects on a variety of memory tasks: New findings and theoretical implications. *Consciousness and Cognition, 5*, 142–164.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustical Society of America, 25*, 975–979.

Cho, Y. S., & Proctor, R. W. (2003). Stimulus and response representations underlying orthogonal stimulus-response compatibility effects. *Psychonomic Bulletin & Review, 10*, 45–73.

Courtire, A., Hardouin, J., Vidal, F., Possamai, C.-A., & Hasbroucq, T. (2003). An additive factor analysis of the effect of sub-anaesthetic doses of nitrous oxide on information processing: Evidence for an impairment of the motor adjustment stage. *Psychopharmacology, 165*, 321–328.

Craik, F. I. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11*, 671–684.

Craik, F. I. M., & Watkins, M. J. (1973). The role of rehearsal in short-term memory. *Journal of Verbal Learning and Verbal Behavior, 12*, 599–607.

De Jong, R. (1993). Multiple bottlenecks in overlapping task performance. *Journal of Experimental Psychology: Human Perception and Performance, 19*, 965–980.

Deutsch, J. A., & Deutsch, D. (1963). Attention: Some theoretical considerations. *Psychological Review, 70*, 80–90.

Donders, F. C. (1969). On the speed of mental processes. In W. G. Koster (Ed.), *Acta Psychologica, 30, Attention and Performance II* (pp. 412–431). Amsterdam: North-Holland. (Original work published in 1868.)

Duncan, J. (1977). Response selection rules in spatial choice reaction tasks. In S. Dornic (Ed.). *Attention and performance VI* (pp. 49–71). Hillsdale, NJ: Erlbaum.

Eberts, R. E. (1994). *User interface design*. Englewood Cliffs, NJ: Prentice-Hall.

Eimer, M. (1998). The lateralized readiness potential as an on-line measure of central response activation processes. *Behavior Research Methods, Instruments, & Computers, 30*, 146–156.

Fitts, P. M. (1951). Engineering psychology and equipment design. In S. S. Stevens (Ed.) *Handbook of experimental psychology* (pp. 1287–1340). New York: Wiley.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*, 381–391.

Fitts, P. M., & Deininger, R. L. (1954). S-R compatibility: Correspondence among paired elements within stimulus and response codes. *Journal of Experimental Psychology, 48*, 483–492.

Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Belmont, CA: Brooks/Cole.

Fitts, P. M., & Seeger, C. M. (1953). S-R compatibility: Spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology, 46*, 199–210.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston: Houghton Mifflin.

Gillan, D. J., Holden, K., Adam, S. Rudisill, M., & Magee, L. (1992). How should Fitts' law be applied to human-computer interaction? *Interacting with Computers, 4*, 291–313.

Godden, D. R., & Baddeley, A. D. (1975). Context-dependent memory in two natural environments: On land and underwater. *British Journal of Psychology, 66*, 325–331.

Grobelny, J., Karwowski, W., & Drury, C. (2005). Usability of graphical icons in the design of human-computer interfaces. *International Journal of Human-Computer Interaction, 18*, 167–182.

Guiard, Y., & Beaudouin-Lafon, M. (2004). Fitts' law 50 years later: Applications and contributions from human-computer interaction. *International Journal of Human-Computer Studies, 61*, 747–750.

Healy, A. F., Wohldmann, E. L., & Bourne, L. E., Jr. (2005). The procedural reinstatement principle: Studies on training, retention, and transfer. In A. F. Healy (Ed.), *Experimental cognitive psychology and its applications* (pp. 59–71). Washington, DC: American Psychological Association.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review, 7*, 185–207.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology, 4*, 11–26.

Hintzman, D. L. (1974). Theoretical implications of the spacing effect. In R. L. Solso (Ed.), *Theories of cognitive psychology: The Loyola symposium* (pp. 77–99). Hillsdale, NJ: Erlbaum.

Hoffman, R. R., & Deffenbacher, K. A. (1992). A brief history of applied cognitive psychology. *Applied Cognitive Psychology, 6*, 1–48.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction*. Cambridge, MA: MIT Press.

Hommel, B., & Prinz, W. (Eds.) (1997). *Theoretical issues in stimulus-response compatibility*. Amsterdam: North-Holland.

Howes, A., & Young, R. M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human-Computer Interaction, 12*, 311–343.

Huettel, S. A., Song, A. W., & McCarthy, G. (2004). *Functional magnetic resonance imaging.* Sunderland, MA: Sinauer Associates.

Hyde, T. S., & Jenkins, J. J. (1973). Recall of words as a function of semantic, graphic, and syntactic orienting tasks. *Journal of Verbal Learning and Verbal Behavior, 12*, 471–480.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology, 45*, 188–196.

Jacko, J. A., & Ward, K. G. (1996). Toward establishing a link between psychomotor task complexity and human information processing. *19th International Conference on Computers and Industrial Engineering, 31*, 533–536.

Jacoby, L. L, & Dallas, M. (1981). On the relationship between autobiographical memory and perceptual learning. *Journal of Experimental Psychology: General, 110*, 306–340.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice Hall.

Kahneman, D., & Tversky, A. (Eds.) (2000). *Choices, values, and frames*. New York: Cambridge University Press.

Keppel, G., & Underwood, B. J. (1962). Proactive inhibition in short-term retention of single items. *Journal of Verbal Learning and Verbal Behavior, 1*, 153–161.

Kiekel, P. A., & Cooke, N. J. (2005). Human factors aspects of team cognition. In R. W. Proctor & K.-P. L. Vu (Eds.), *Handbook of human factors in web design* (pp. 90–103). Mahwah, NJ: Erlbaum.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391–438.

Klein, R. M., & Shore, D. I. (2000). Relation among modes of visual orienting. In S. Monsell, & J. Driver (Eds.), *Control of cognitive processes: Attention and performance XVIII* (pp. 195–208). Cambridge, MA: MIT Press.

Kornblum, S., Hasbroucq, T., & Osman, A. (1990). Dimensional overlap: Cognitive basis for stimulus-response compatibility—A model and taxonomy. *Psychological Review, 97*, 253–270.

Lachman, R., Lachman, J. L., & Butterfield, E. C. (1979). *Cognitive psychology and information processing: An introduction*. Hillsdale, NJ: Erlbaum.

Lavie, N., Hirst, A., de Fockert, J. W., & Viding, E. (2004). Load theory of selective attention and cognitive control. *Journal of Experimental Psychology: General, 133*, 339–354.

Lee, E., & MacGregor, J. (1985). Minimizing user search time in menu retrieval systems. *Human Factors, 27*, 157–162.

Lockhart, R. S., & Murdock, B. B., Jr. (1970). Memory and the theory of signal detection. *Psychological Bulletin, 74*, 100–109.

Loftus, E. F., & Palmer, J. C. (1974). Reconstruction of automobile destruction: An example of the interaction between language and memory. *Journal of Experimental Psychology: Human Learning and Memory, 4*, 19–41.

Lu, C.-H., & Proctor, R. W. (1995). The influence of irrelevant location information on performance: A review of the Simon effect and spatial Stroop effects. *Psychonomic Bulletin & Review, 2*, 174–207.

MacKenzie, I. S., & Soukoreff, R. W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction, 17*, 147–198.

Mackworth, N. H. (1948). The breakdown of vigilance during prolonged visual search. *Quarterly Journal of Experimental Psychology, 1*, 6–21.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide* (2nd ed.). Mahwah, NJ: Erlbaum.

McClelland, J. L. (1979). On the time relations of mental processes: A framework for analyzing processes in cascade. *Psychological Review, 88*, 375–407.

McDonald, J. E., Stone, J. D., & Liebelt, L. S. (1983). Searching for items in menus: The effects of organization and type of target. *Proceedings of the Human Factors Society 27th Annual Meeting* (pp. 289–338). Hillsdale, NJ: Erlbaum.

Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Smith, J. E. K. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review, 86*, 340–370.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory-period phenomena. *Psychological Review, 104*, 749–791.

Michaels, C. F., & Stins, J. F. (1997). An ecological approach to stimulus-response compatibility. In B. Hommel & W. Prinz (Eds.), *Theoretical issues in stimulus-response compatibility* (pp. 333–360). Amsterdam: North-Holland.

Miller, G. A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81–97.

Miller, J. (1988). Discrete and continuous models of human information processing: Theoretical distinctions and empirical results. *Acta Psychologica, 67*, 191–257.

MIT Press (2007). The Journal of Cognitive Neuroscience Instructions for Authors. Retrieved February 20, 2007, from http://jocn.mitpress.org/misc/ifora.shtml.

Monsell, S., & Driver, J. (Eds.) (2000). *Control of cognitive processes: Attention and performance XVIII*. Cambridge, MA: MIT Press.

Morris, C. D., Bransford, J. D., & Franks, J. J. (1977). Levels of processing versus transfer appropriate processing. *Journal of Verbal Learning and Verbal Behavior, 16*, 519–533.

Nairne, J. S. (2003). Sensory and working memory. In A. F. Healy & R. W. Proctor (Eds.), *Experimental psychology*. Volume 4 of *Handbook of psychology*, Editor-in-Chief: I. B. Weiner. Hoboken, NJ: Wiley.

Navon, D. (1984). Resources—a theoretical soup stone? *Psychological Review, 91*, 216–234.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1–55). Hillsdale, NJ: Erlbaum.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Oulasvirta, A., Kärkkäinen, L., & Laarni, J. (2005). Expectations and memory in link search. *Computers in Human Behavior, 21*, 773–789.

Oulasvirta, A., & Saariluoma, P. (2004). Long-term working memory and interrupting messages in human-computer interaction. *Behaviour & Information Technology, 23*, 53–64.

Pachella, R. G. (1974). The interpretation of reaction time in information-processing research. In B. H. Kantowitz (Ed), *Human information processing: Tutorials in performance and cognition* (pp. 41–82). Hillsdale, NJ: Erlbaum.

Parasuraman, R. (2003). Neuroergonomics: Research and practice. *Theoretical Issues in Ergonomics Science, 4*, 5–20.

Parasuraman, R., & Davies, D. R. (1977). A taxonomic analysis of vigilance performance. In R. R. Mackie (Ed.), *Vigilance: Theory, operational performance, and physiological correlates* (pp. 559–574). New York: Plemum.

Parsons, M. W., Harrington, D. L., & Rao, S. M. (2005). Distinct neural systems underlie learning visuomotor and spatial representations of motor skills. *Human Brain Mapping, 24*, 229–247.

Pashler, H. (1998). *The psychology of attention*. Cambridge, MA: MIT Press.

Payne, S. J. (1995). Naïve judgments of stimulus-response compatibility. *Human Factors, 37*, 495–506.

Percival, L. C., & Noonan, T. K. (1987). Computer network operation: Applicability of the vigilance paradigm to key tasks. *Human Factors, 29*, 685–694.

Perlman, G. (1984). Making the right choices with menus. *Proceedings of INTERACT '84* (pp. 291–295). London: IFIP.

Peterson, L. R., & Peterson, M. J. (1959). Short-term retention of individual verbal items. *Journal of Experimental Psychology, 58*, 193–198.

Posner, M. I. (1986). Overview. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.). *Handbook of perception and human performance vol. II: Cognitive processes and performance* (pp. V3–V10). New York: Wiley.

Posner, M. I., & Cohen, Y. (1984). Components of visual orienting. In H. Bouma & D. G. Bouwhuis (Eds.) *Attention and performance X* (pp. 531–556). Hillsdale, NJ: Erlbaum.

Proctor, R. W., & Van Zandt, T. (1994). *Human factors in simple and complex systems*. Boston: Allyn & Bacon.

Proctor, R. W., & Vu, K.-P. L. (2006). *Stimulus-response compatibility: Data, theory and application*. Boca Raton, FL: CRC Press.

Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review, 111*, 333–367.

Reeve, T. G., & Proctor, R. W. (1990). The salient features coding principle for spatial- and symbolic-compatibility effects. In R. W. Proctor & T. G. Reeve (Eds.) *Stimulus-response compatibility: An integrated perspective* (pp. 163–180). Amsterdam: North-Holland.

Roediger, H. L., III, & McDermott, K. B. (1995). Creating false memories: Remembering words not presented in lists. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*, 803–814.

Roscoe, S. (2005). Historical overview of human factors and ergonomics. In R. W. Proctor & K.-P. L. Vu (Eds.), *Handbook of human factors in web design* (pp. 3–12). Mahwah, NJ: Erlbaum.

Rubichi, S., Vu, K.-P. L., Nicoletti, R., & Proctor, R. W. (2006). Spatial coding in two dimensions. *Psychonomic Bulletin & Review, 13*, 201–216..

Rugg, M. D., & Coles, M. G. H. (Eds.) (1995). *Electrophysiology of mind: Event-related brain potentials and cognition*. Oxford, England: Oxford University Press.

Salthouse, T. A. (2005). From description to explanation in cognitive aging. In R. J. Sternberg & J. E. Pretz (Eds.), *Cognition & intelligence: Identifying the mechanisms of the mind* (pp. 288–305). Cambridge, England: Cambridge University Press.

Sanders, A. F. (1998). *Elements of human performance*. Mahwah, NJ: Erlbaum.

Schmorrow, D. D. (Ed.) (2005). *Foundations of augmented cognition*. Mahwah, NJ: Erlbaum.

Schvaneveldt, R. W. (2005). Finding meaning in psychology. In A. F. Healy (Ed.), Experimental cognitive psychology and its applications (pp. 211–235). Washington, DC: American Psychological Association.

Schweickert, R. (1978). A critical path generalization of the additive factor method: Analysis of a Stroop task. *Journal of Mathematical Psychology, 18*, 105–139.

Schweickert, R., & Boruff, B. (1986). Short-term memory capacity: Magic number or magic spell? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12*, 419–425.

See, J. E., Howe, S. R., Warm, J. S., & Dember, W. N. (1995). Meta-analysis of the sensitivity decrement in vigilance. *Psychological Bulletin, 117*, 230–249.

Shaffer, L. H. (1965). Choice reaction with variable S-R mapping. *Journal of Experimental Psychology, 70*, 284–288.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science, 171*, 701–703.

Simon, J. R. (1990). The effects of an irrelevant directional cue on human information processing. In R. W. Proctor & T. G. Reeve (Eds.), *Stimulus-response compatibility: An integrated perspective* (pp. 31–86). Amsterdam: North-Holland.

Slamecka, N. J. (1968). An examination of trace storage in free recall. *Journal of Experimental Psychology, 76*, 504–513.

Slamecka, N. J., & Graf, P. (1978). The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human Learning and Memory, 4*, 592–604.

Smith, T. J., & Henning, R. A. (2005). Cybernetics of augmented cognition as an alternative to information processing. In D. D. Schmorrow (Ed.), *Foundations of augmented cognition* (pp. 641–650). Mahwah, NJ: Erlbaum.

Somberg, B. L., & Picardi, M. C. (1983). Locus of information familiarity effect in search of computer menus. *Proceedings of the Human Factors Society 27th Annual Meeting* (pp. 826–830). Santa Monica, CA: Human Factors Society.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders' method. In W. G. Koster (Ed), *Attention and Performance II. Acta Psychologica, 30*, 276–315.

Sternberg, S., Monsell, S, Knoll, R. L., & Wright, C. E. (1978). The latency and duration of rapid movement sequences. In G. E. Stelmach (Ed.), *Information processing in motor control and learning*. New York: Academic Press.

Talmi, D., Grady, C. L., Goshen-Gottstein, Y., & Moscovitch, M. (2005). Neuroimaging the serial position curve: A test of single-store versus dual-store models. *Psychological Science, 16*, 716–723.

Teichner, W. H., & Krebs, M. J. (1974). Laws of visual choice reaction time. *Psychological Review, 81*, 75–98.

Tlauka, M. (2004). Display-control compatibility: The relationship between performance and judgments of performance. *Ergonomics, 47*, 281–295.

Tombu, M., & Jolicœur, P. (2005). Testing the predictions of the central capacity sharing model. *Journal of Experimental Psychology: Human Perception and Performance, 31*, 790–802.

Treisman, A. M. (1964). Selective attention in man. *British Medical Bulletin, 20*, 12–16.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12*, 97–136.

Trimmel, M., & Huber, R. (1998). After-effects of human-computer interaction indicated by P300 of the event-related brain potential. *Ergonomics, 41*, 649–655.

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review, 80*, 359–380.

Tullis, T. S., Catani, M., Chadwick-Dias, A., & Cianchette, C. (2005). Presentation of information. In R. W. Proctor & K.-P. L. Vu (Eds.), *Handbook of human factors in web design* (pp. 107–133). Mahwah, NJ: Erlbaum.

Usher, M., Olami, Z., & McClelland, J. L. (2002). Hick's law in a stochastic race model with speed-accuracy tradeoff. *Journal of Mathematical Psychology, 46*, 704–715.

Valle-Inclán, F., de Labra, C., & Redondo, M. (2000). Psychophysiological studies of unattended information processing. *The Spanish Journal of Psychology, 3*, 76–85.

Van Zandt, T., Colonius, H., & Proctor, R. W. (2000). A comparison of two response time models applied to perceptual matching. *Psychonomic Bulletin & Review, 7*, 208–256.

Verhaeghen, P., & Marcoen, A. (1996). On the mechanisms of plasticity in young and older adults after instruction in the method of loci: Evidence for an amplification model. *Psychology & Aging, 11*, 164–178.

Vu, K.-P. L., Bhargav, A., & Proctor, R. W. (2003). Imposing password restrictions for multiple accounts: Impact on generation and recall of passwords. *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 1331–1335). Santa Monica, CA: HFES.

Vu, K.-P. L, & Proctor, R. W. (2003). Naïve and experienced judgments of stimulus-response compatibility: Implications for interface design. *Ergonomics, 46*, 169–187.

Vu, K.-P. L, & Proctor, R. W. (2004). Mixing compatible and incompatible mappings: Elimination, reduction, and enhancement of spatial compatibility effects. *Quarterly Journal of Experimental Psychology, 57A*, 539–556.

Vu, K.-P. L., Tai, B.-L., Bhargav, A., Schultz, E. E., & Proctor, R. W. (2004). Promoting memorability and security of passwords through sentence generation. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (pp. 1478–1482). Santa Monica, CA: HFES.

Walker, N., Smelcer, J. B., & Nilsen, E. (1991). Optimizing speed and accuracy of menu selection: A comparison of walking and pull-down menus. *International Journal of Man-Machine Studies, 35*, 871–890.

Wang, H., & Proctor, R. W. (1996). Stimulus-response compatibility as a function of stimulus code and response modality. *Journal of Experimental Psychology: Human Perception and Performance, 22*, 1201–1217.

Wickelgren, W. A. (1977). Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica, 41*, 67–85.

Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman and D. R. Davies (Eds.), *Varieties of attention* (pp. 63–102). San Diego, CA: Academic Press.

Wickens, C. D., Lee, J., Liu, Y., & Becker, S. (2004). *An introduction to human factors engineering* (2nd ed.). New York: Prentice Hall.

Wickens, D. D. (1970). Encoding categories of words: An empirical approach to meaning. *Psychological Review, 77*, 1–15.

Wolfe, J. M. (1994). Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1*, 202–238.

Wood, L. E., & Pratt, J. D. (1987). Pegword mnemonic as an aid to memory in the elderly: A comparison of four age groups. *Educational Gerontology, 13*, 325–339.

Yantis, S. (2000). Goal-directed and stimulus-driven determinants of attentional control. In S. Monsell, & J. Driver (Eds.), *Control of cognitive processes: Attention and performance XVIII* (pp. 195–208). Cambridge, MA: MIT Press.

Young, P. M., Clegg, B. A., & Smith, C. A. P. (2004). Dynamic models of augmented cognition. *International Journal of Human-Computer Interaction, 17*, 259–273.

·3·

MENTAL MODELS IN HUMAN-COMPUTER INTERACTION

Stephen J. Payne
University of Manchester

What Is a Mental Model? **64**
Mental Models of Text and Other Artifacts **67**
Instructions for Mental Models **69**
Multimedia Instruction 69
The Theory of Learning by Not Doing 70
Shared Mental Models **72**
References **74**

The plan for this chapter is as follows It begins by reviewing and discussing the term "mental models" as it has been used in the literature on human-computer interaction (HCI), and in the neighboring disciplines of cognitive psychology where it was first coined. There is little consensus on what exactly is and is not a mental model, and yet it is too widely used for any posthoc attempt at a narrower definition to somehow cleanse the field. In consequence, I characterize several layers of theoretical commitment that the term may embrace, following an earlier discussion (Payne, 2003). To illustrate the argument, several classic and more recent studies from the HCI literature will be reviewed, with pointers to others. This first part of the chapter is based on material published in Payne (2003).

In cognitive psychology, mental models have major currency in two sub-disciplines—text comprehension and reasoning, although in the former they more often currently go by the name "situation models." Discussion in the latter focuses on quite refined theoretical disputes that currently have little relevance for HCI. The work on text comprehension, however, is germane. With the advent of the Web, the comprehension of text of various kinds has become a dominant mode of HCI, with important design issues for Websites, digital libraries, and so on. Interaction with text is in some ways a paradigm for interaction with information. With these points in mind, the concept of mental models in text comprehension will be discussed, with a particular eye to the issues that HCI accentuates, such as understanding multiple texts.

Two of the major practical questions raised by mental models are (a) how are they acquired and how can their acquisition be supported by instruction? The third section of this chapter will discuss two angles on these questions in HCI: first, the use of interactive computation and multimedia as an instructional method; second, the important tension between exploration and instruction, first systematically discussed in the HCI literature by Carroll's (1990) work on minimalism.

Finally, the paper will review some recent work on the importance of mental models for understanding aspects of collaborative teamwork. This area suggests that a relatively expansive view of human knowledge representations may be necessary for progress in HCI.

Throughout the chapter, a particular approach is taken to review: to choose one or two key studies and report them in some detail. I hope that this will allow some of the empirical methodologies and the rich variation in these to be conveyed. The chosen studies will be accompanied by some further references to the literature, but there are too many subtopics reviewed to aim for completeness.

WHAT IS A MENTAL MODEL?

The user's mental model of the device is one of the more widely discussed theoretical constructs in HCI. Alongside wide-ranging research literature, even commercial style guides have appealed to mental models for guidance (i.e., Mayhew, 1992; Tognazzini, 1992; Apple Human Interface Guidelines Apple Computer Inc., 1987).

Yet a casual inspection of the HCI literature reveals that mental models are used to label many different aspects of users' knowledge about the systems they use. Nevertheless, I propose that even this simple core construct—what users know and believe about the systems they use—is worth highlighting and promoting. It is more distinctive than it might first seem, especially in comparison with other cognitive-science approaches. Further, beyond the core idea there is a progression of stronger theoretical commitments that have been mobilized by the mental models label, each of which speaks to important issues in HCI research, if not yet in practice.

The fundamental idea is that the *contents* of people's knowledge, including their theories and beliefs, can be an important explanatory concept for understanding users' behavior in relation to systems. This idea may seem obvious and straightforward, but in fact it suggests research questions that go against the grain of most contemporary cognitive psychology, which has concerned itself much more with the general limits of the human-information-processing system, such as the constraints on attention, retrieval, and processing. Thus, cognitive psychology tends to focus on the *structure* of the mind, rather than its contents. (The major exception to the rule that cognitive psychology has been obsessed with architecture over content is the work on expertise, and even here, recent work has focused on explanations of extreme performance in terms of general independent variables such as "motivated practice," i.e., Ericsson, Krampe, and Tesch-Romer (1993), rather than epistemological analysis.)

Refocusing attention on mental content about particular domains is what made mental models a popular idea in the early 1980s, such as the papers in Gentner and Stevens (1983). For example, work on naïve physics (i.e., McClosky, 1983) attempts to explain people's reasoning about the physical world, not in terms of working memory limits or particular representations, but in terms of their beliefs about the world, such as the nature of their theories of mechanics or electricity, for example. This focus on people's knowledge, theories, and beliefs about particular domains transfers naturally to questions in HCI, where practical interest may focus on how users conceive the workings of a particular device; how their beliefs shape their interactive behavior, and what lessons may be drawn for design.

In this mold, consider a very simple study of my own (Payne, 1991). Students were interviewed about ATMs. Following Collins and Gentner (1987) among others, "what if" questions were posed to uncover student's theories about the design and function of ATMs. For example, students were asked whether machines sometimes took longer to process their interactions; what information was stored on the plastic card; and what would happen if they "typed ahead" without waiting for the next machine prompt.

The interviews uncovered a wide variety in student's beliefs about the design of ATMs. For example, some assumed that the plastic card was written to as well as read from during transactions, and thus could encode the current balance of their account. Others assumed that the only information on the card was the user's personal identification number, allowing the machine to check the identity of the user (as it turns out, both these beliefs are incorrect). A conclusion from this simple observation is that users of machines are eager to form explanatory models and will readily go beyond available data to infer models that are consistent with their experiences. (One might

wonder whether such explanations were not merely ad hoc, prompted during the interview: in fact some were, but explicit linguistic cues—such as "I've always thought"—strongly suggested that many were not.)

Another observation concerning students "models" of ATMs was that they were fragmentary, perhaps more fragmentary than the term "model" might ordinarily connote: they were collections of beliefs about parts of the system, processes, or behaviors, rather than unified models of the whole design. Students would happily recruit an analogy to explain one part of the machine's operation that bore no relation to the rest of the system. This fragmentary character of mental models of complex systems may be an important aspect (see i.e., Norman, 1983), allowing partial understandings to be maintained. One implication is that users' mental models of single processes or operations might be a worthwhile topic for study and practical intervention (in design or instruction).

One widely held belief about a particular process affected the students' behavior as users. Almost all respondents believed that it was not possible to type ahead during machine pauses. At the time the study was conducted this was true for some, but not all, designs in use. Consequently, in some cases transactions were presumably being needlessly slowed because of an aspect of users' mental models.

A more recent study of a similar kind is an investigation of users' models of the navigation facilities provided by Internet browsers (Cockburn & Jones, 1996). Internet browsers, like Internet Explorer, maintain history lists of recently visited pages, providing direct access to these pages without needing to enter the URL or follow a hyperlink. The "back" and "forward" buttons provide a very frequently used mechanism for browsing history lists, but do users have good mental models for how they work? Cockburn and Jones (1996) showed that many do not.

The history list of visited pages can be thought of as a stack: a simple last-in-first-out data structure to which elements can be added (pushed) or taken out (popped) only from the top (consider a stack of trays in a canteen). When a new web page is visited by following a hyperlink, or by entering a URL, its address is pushed onto the top of the stack. This is true even if the page is already in the history list, so that the history list may contain more than one copy of the same page. However, when a page is visited by using the Back button (or, at least typically, by choosing from the history list), the page is not pushed onto the stack. So, what happens when the currently displayed page is not at the top of the stack (because it has been visited via the history list) and a new link is followed (or a new URL entered)? The answer is that all the pages in the history list that were above the current page are popped from the stack, and the newly visited page is pushed onto the stack in their place. For this reason the history list does *not* represent a complete record, or time-line of visited pages, and not all pages in the current browsing episode can be backed-up to. In Cockburn and Jones' study, few users appreciated this aspect of the device.

This then, has been the major thrust of work on mental models in HCI: what do people know and believe to be true about the way the systems they interact with are structured? How do their beliefs affect their behavior? In this literature a "mental model" is little more than a pointer to the relevant parts of the user's knowledge, yet this is not to deny its usefulness. One approach that it has engendered is a typology of knowledge—making groupings and distinctions about types of knowledge that are relevant in certain circumstances. It is in exactly this way that a literature on "shared mental models" as an explanatory concept in teamwork has been developed. This topic is perhaps the most rapidly growing area of mental models research in HCI and will be reviewed in the final section of this chapter.

However, as argued at length in Payne (2003), there are approaches to mental models in HCI that go beyond a concern with user knowledge and beliefs to ask more nuanced theoretical questions. The first of these is to investigate the form of mental models by inspecting the processes through which mental models might have their effects on behavior.

A powerful idea here is that mental models of machines provide a problem space that allows more elaborate encoding of remembered methods, and in which novice or expert problem solvers can search for new methods to achieve tasks.

The classic example of this approach is the work of Halasz and Moran (1983) on Reverse Polish Notation (RPN) calculators. RPN is a post-fix notation for arithmetic, so that to express 3 + 4, one would write 3 4 +. RPN does away with the need for parentheses to disambiguate composed operations. For example (1 + 2) * 3 can be expressed 1 2 + 3 * with no ambiguity. RPN calculators need a key to act as a separator between operands, which is conventionally labeled ENTER, but they do not need an = key, as the current total can be computed and displayed whenever an operator is entered.

Halasz and Moran taught one group of students how to use an RPN calculator using instructions, like a more elaborate version of the introduction above, which simply described the appropriate syntax for arithmetic expressions. A second group of subjects was instructed, using a diagram, about the stack model that underlies RPN calculation. Briefly, when a number is keyed in, it is "pushed" on top of a stack-data structure (and the top slot is displayed). The ENTER key copies the contents of the top slot down to the next slot. Any binary arithmetic operation is always performed on the contents of the top two slots and leads to the result being in the top slot, with the contents of slots 3 and below moving up the stack.

Halasz and Moran discovered that the stack-model instructions made no difference to participants' ability to solve routine arithmetic tasks: the syntactic "method-based" instructions sufficed to allow participants to transform the tasks into RPN notation. However, for more creative problems (such as calculating (6 + 4) and (6 + 3) and (6 + 2) and only keying the number 6 once) the stack group was substantially better. Verbal protocols showed that these subjects reasoned about such problems by mentally stepping through the transformations to the stack at each keystroke.

This kind of reasoning, stepping through a sequence of states in some mental model of a machine, is often called "mental simulation" in the mental models literature, and the kind of model that allows simulation is often called a "surrogate" (Young, 1983; Carroll & Olson, 1988). From a practical standpoint, the key property of this kind of reasoning is that it results in behavior that is richer and more flexible than the mere rote following of learned methods. The idea that the same method may be encoded more richly, so that it is more flexible

and less prone to forgetting will be returned to later in the chapter when a theory of mental models of interactive artifacts is considered, and when ideas about instruction for mental models are reviewed.

A second example of mental models providing a problem space elaboration of rote methods comes in the work of Kieras and Bovair (1984). This research was similar to that of Halasz and Moran (1983) in that it compared the learning performance of two groups: (a) one instructed with rote procedures, (b) the other additionally with a diagrammatic model of the device on which the procedures were enacted. In this case, the device was a simple control panel, in which each rote procedure specified a sequence of button-pushes and knob-positions leading to a sequence of light-illuminations. The model was a circuit diagram showing the connections between power-source switches and display-lights.

Kieras and Bovair (1984) found that the participants instructed with the model learned the procedures faster, retained the procedures more accurately, executed the procedures faster, and could simplify inefficient procedures that contained redundant switch settings. They argued that this was because the model (circuit diagram) explained the contingencies in the rote-action sequences (i.e., if a switch is set to MA, so that the main accumulator circuit is selected, then the FM, fire main, button must be used).

A related theoretical idea is that mental models are a special kind of representation, sometimes called an *analog* representation: one that shares the structure of the world it represents. This was taken as the definitional property of mental models by the modern originator of the term, the British psychologist Kenneth Craik (1943). It is this intuition that encourages the use of terms like "mental simulation"—the intuition that a mental model is like a physical model, approximating the structure of what it represents, just as a model train incorporates (aspects of) the physical structure of a train.

The idea that mental models are analog in this sense is a definitional property in the work on reasoning and comprehension by Johnson-Laird (Johnson-Laird, 1983, 1989; this will be further discussed in part 2, concerning representations of text) and also in the theory of Holland, Holyoak, Nisbett, and Thagard (1986) and Moray (1999). However, there are different nuances to the claim, which must be considered. And, in addition, there is a vexed question to be asked; namely, what is the explanatory or predictive force of a commitment to analog representational form? Is there any reason for HCI researchers to pay attention to theoretical questions at this level?

Certainly, this is the view of Moray (1999) who is concerned with mental models of complex dynamic systems, such as industrial plants. He proposes that models of such systems are structure-sharing *homomorphisms* rather than isomorphisms, i.e. they are many to one rather than one-to-one mappings of objects, properties, and relations. (In this he follows Holland, Holyoak, Nisbett, & Thagard, 1986).

Homomorphic models of dynamic systems may not share structure with the system at the level of static relations, but only at the level of state-changes. Thus, such models have the character of state-transition diagrams, making the empirical consequences of structure sharing somewhat unclear, because any problem space can be represented in this way.

In my view, a clearer view of the explanatory force of analog mental models can be derived by carefully considering the ideas of computational and informational equivalence first introduced by Simon (1978).

It is obviously possible to have two or more distinct representations of the same information. Call such representations "informationally equivalent" if all the information in one is inferable from the other, and vice versa. Two informationally equivalent representations may or may not additionally be "computationally equivalent," meaning that the cost structure of accessing and processing the information is equivalent in both cases, or, as Larkin and Simon (1987) put it: "information given explicitly in the one can also be drawn easily and quickly from the information given explicitly in the other, and vice versa." As Larkin and Simon point out, "easily" and "quickly" are not precise terms, and so this definition of computational equivalence is inherently somewhat vague; nevertheless it points to empirical consequences of a representation (together with the processes that operate upon it) that depend on form, and therefore go beyond mere informational content.

In Payne (2003), I propose adopting *task-relative* versions of the concepts of informational and computational equivalence. Thus, representations are informationally equivalent, *with respect to a set of tasks*, if they allow the same tasks to be performed (i.e. contain the requisite information for those tasks). The representations are, additionally, computationally equivalent with respect to the tasks they allow to be performed, if the *relative difficulty* of the tasks is the same, whichever representation is being used. (Note that according to these definitions, two representations might be computationally equivalent with regard to a subset of the tasks they support but not with regard to the total set, so that in Larkin and Simon's sense they would merely be informationally equivalent. The task-relative versions of the constructs thus allow more finely graded comparisons between representations.)

This idea can express what is behaviorally important about the idea of analog models, or structure-sharing mental representations of a state of affairs of a dynamic system. An analog representation is computationally equivalent (with respect to some tasks) to external perception and manipulation of the state of affairs it represents.

Bibby and Payne (1993; 1996) exploited this distinction between computational and informational equivalence in the domain of HCI, using a computer simulation of a device derived from that studied by Kieras and Bovair (1984). The device was a multiroute circuit, in which setting switches into one of several configurations would make a laser fire; various indicator lights showed which components of the circuit were receiving power. What concerned Bibby and Payne (1993) was the idea of computational equivalence between a mental model and a diagram of the device, rather than the device itself.

Bibby and Payne asked participants to repeatedly perform two types of tasks: a switch task, in which all but one switch was already in position to make a laser fire (the participant had to key the final switch) and a fault task, in which the pattern of indicator lights was such that one of the components must be broken (the participant had to key the name of the broken component).

Participants were instructed about the device with either a table, which showed the conditions under which each indica-

tor light would be illuminated, or with procedures, sequences of switch positions enabling the laser to be fired. Both instructions were sufficient for both switch and fault tasks; they were informationally equivalent with respect to those tasks. However, the table made the fault task easier than the switch task, whereas the procedures made the switch task easier.

During practice, when participants consulted the instructions, this pattern of relative difficulty was confirmed by a crossover interaction in response times. Furthermore, when the instructions were removed from the participants, so that they had to rely on their mental representation of the device, the crossover interaction persevered, demonstrating that the mental representations were computationally equivalent to the external instructions.

In subsequent experiments, Bibby and Payne (1996) demonstrated that this pattern persevered even after considerable interaction with the device that might have been expected to provide an opportunity to overcome the representational constraints of the initial instruction. The crossover interaction eventually disappeared only after extended practice on the particular fault-and-switch task (80 examples of each: perhaps because of asymptotic performance having been reached). At this point, Bibby and Payne introduced two similar but new types of tasks designed so that once again, the table favored one task whereas procedures favored the other. (However, the device instructions were *not* re-presented.) At this point the crossover reappeared, demonstrating that participants were consulting their instructionally derived mental model of the device, and that this was still in a form computationally equivalent to the original external representation of the instructions.

Practically, this research shows that the exact form of instructions may exert long-lasting effects on the strategies that are used to perform tasks, so that designers of such instructions must be sensitive not only to their informational content but also to their computational properties. In this light, they also suggest that one instructional representation of a device is very unlikely to be an optimal vehicle for supporting all user tasks: it may well be better to provide different representations of the same information, each tailored to particular tasks. In this sense, perhaps instructions should mirror and exploit the natural tendency, noted above, for users to form fragmentary mental models, with different fragments for different purposes.

In terms of theory, Bibby and Payne's findings lend support to the suggestion developed above that mental models of a device that are formed from instructions may be computationally equivalent to the external representations of the device. This idea gives a rather new reading, and one with more ready empirical consequences to the theoretically strong position that mental models are essentially analog, homomorphic representations.

MENTAL MODELS OF TEXT AND OTHER ARTIFACTS

The psychological literature on text comprehension has been transformed by the idea of a situation model, first put forward as part of a general theory of text comprehension by van Dijk and Kintsch (1983), and developed over the years by Kintsch (e.g., 1998) and followers. The central idea of the general theory is that readers construct mental representations of what they read at several different levels. First, they encode the surface form of the text: the words and syntax. Second, they go beyond this to a representation of the propositional content of the text. Finally, they go beyond the propositional context of the text itself to represent what the text is about, incorporating their world knowledge to construct a situation model or mental model of the described situation.

(Under this view, it is the content that distinguishes a situation model from a text base, rather than a representational format. However, some researchers, notably Johnson-Laird (1983), and followers have pursued the idea of mental models derived from text as analog representations of the described situation. Thus, in text comprehension, there is a version of the issue discussed in part one.)

It is instructive to consider some of the evidence for situation models, and what important issues in text comprehension the theory of situation models allows us to address.

A classic early study was conducted by Bransford, Barclay, and Franks (1972). They asked participants to read simple sentences such as,

Three turtles rested beside/on a floating log, and a fish swam beneath them.

(The slash indicates that some subjects read the sentence with the word "beside", and others read the same sentence with the word "on").

In a later recognition test, interest centered on how likely readers were to falsely accept minor rewordings of the original sentences. In the above case, the foil sentence was

Three turtles rested beside/on a floating log, and a fish swam beneath it.

The key finding was that people who had read the "on" versions of the sentences were much more likely to accept the changed version of the sentence, despite the fact that that at the level of the sentences the difference between original and foil sentences in the two conditions is identical, limited in each case to the last word of the sentence. The reason for false recognition in one case is because, in this case, but not when "on" is replaced by "beside," the original and foil sentences describe the same situation.

A related series of experiments was reported by Fletcher and Chrysler (1990). In a series of carefully controlled experiments, they varied the overlap between sentences in a recognition test and sentences from 10 different texts read by the participants. Each text described a state of affairs (i.e., the relative cost of antiques) consistent with a linear ordering among a set of five objects. They found that participants were influenced by overlap between sentences at study and test corresponding to the three levels of discourse representation proposed by van Dijk and Kintsch (1983): surface form, text base, and situation model. Recognition performance was best when distracter items were inconsistent with all three levels of representation. Recognition was above chance when distracters violated merely the surface form of the original sentences (i.e. substituting rug for carpet). It improved further when propositional information from the

text base, but not the linear ordering of the situation, was violated. Recognition was best of all when the distracters were inconsistent with the situation described by the text. This suggests that the some aspects of the structure of the situation (in this case a set of linear orderings) were retained.

Next, consider work by Radvansky and Zacks (Radvansky & Zacks, 1991; Radvansky, Spieler, & Zacks, 1993). In these experiments, participants read sentences such as, "The cola machine is in the hotel," each of which specified the location of an object. In one condition sentences shared a common object (i.e. cola machine) but different locations. In a second condition, different objects share a common location (i.e. the city hall). Later in the experiment participants were given a speeded-recognition test. Radvansky and Zacks found a significant fan effect (Anderson, 1974) for the common object condition; times to verify sentences increased as the number of different locations rose. For the common location sentences no significant fan effect emerged. This was interpreted as evidence that participants formed mental models around the common location (a representation of such a location containing all the specified objects) and retrieval from long-term memory was organized around these mental models. It is impossible, or much harder, to form such a representation of the same object in multiple locations.

What all these studies, and many like them, reveal is that when understanding text, readers spontaneously construct a mental representation that goes beyond the text itself, and what it means and use inferences to construct a richer model of what the text is about—a situation model.

Beyond these refined and clever, but undeniably rather narrow experimental contexts, the construct of situation model has been put to work to illuminate some practical issues concerning text comprehension, and exactly this issue will be returned to later, where we will see how it can inform attempts to understand instructional strategies for engendering useful mental models.

There are two principle ways in which the literature on text comprehension is relevant to HCI. First, it provides support for the idea that a mental model is a representation of what a representational artifact represents. The layered model of text comprehension previously outlined can be generalized to the claim that the user of any representational artifact must construct a representation of the artifact itself, and of what the artifact represents, and of the mapping between the two (how the artifact represents). This is the basis of the Yoked State Space (YSS) hypothesis (Payne, Squibb, & Howes, 1990).

If a reader's goal is just to understand a text, as it was in the experiments just reviewed, then the text-representation can be discarded once a model has been constructed. However, there are many tasks of text *use,* in which it is necessary to maintain a representation of the text, alongside a mental model of the meaning of the text. Consider, for example, the tasks of writing and editing, or of searching for particular content in a text. In such tasks, it is necessary to keep in mind the relation between the surface form of the text—wording, spatial layout, etc.—and its meaning. Text is a representational artifact, and to *use* it in this sense one needs a mental representation of the structure of the text, and of the "situation" described by the text and of the mapping between the two.

According to the Yoked State Space hypothesis (Payne, Squibb, & Howes, 1990), this requirement is general to all representational artifacts, including computer systems. To use such artifacts requires some representation of the domain of application of the artifact—the concepts the artifact allows you to represent and process. The user's goals are states in this domain, which is therefore called the goal space. However, states in the goal space cannot be manipulated directly. Instead, the user interacts with the artifact, and therefore needs knowledge of the artifact, and of the operations that allow states of the artifact to be transformed. Call this problem space the device space. In order to solve problems in the goal space by searching in the device space, the user must know how the device space represents the goal space. In this sense the two spaces need to be yoked. The minimal device space for a certain set of tasks must be capable of representing all the states in the corresponding goal space. More elaborate device spaces may incorporate device states that do not directly represent goal states, but which allow more efficient performance of tasks, just as the stack model of an RPN calculator allows an elaboration of methods for simple arithmetic.

The work of Halasz and Moran (1983) can readily be assimilated into the YSS framework. The no-model condition was provided with enough information to translate algebraic expressions into their Reverse Polish equivalent. However, in this understanding of RP expressions, the ENTER key was given merely an operational account, serving simply as a separator of operands, and did not transform the device state. The stack model, however, provides a figurative account of the ENTER key.

This discussion illustrates a practical lesson for the design of interfaces and instructions. In the case of the copy buffer and the calculator stack, the standard interface does not allow the appropriate device space readily to be induced, so that conceptual instructions must fill the gap. The obvious alternative, which has been developed to some extent in both cases, is to redesign the user interface so as to make the appropriate device space visible. These examples suggest a simple heuristic for the provision of conceptual instructions that may help overcome the considerable controversy over whether or not such instructions (as opposed to simple procedural instructions) are useful (see, i.e., Wright, 1988). According to this heuristic, conceptual instructions will be useful if they support construction of a YSS that the user would otherwise have difficulty inducing (Payne, Howes, & Hill, 1992).

A more direct way in which text comprehension research is relevant to HCI is that so much HCI is reading text. Beyond the standard issues, the widespread availability of electronic texts raises some new concerns that have not yet seen much work, yet are perhaps the most directly relevant to HCI design. Two issues stand out: (a) the usability of documents that incorporate multiple media alongside text, and (b) the exploitation by readers of multiple texts on the same topic.

How are multimedia "texts" that incorporate graphics comprehended? There is only a small literature on this within the mainstream field of text comprehension, but this literature exploits the idea of a mental model.

Glenberg and Langston (1992) argued that the widespread idea that diagrams can assist the comprehension of technical

text had, at the time, been little tested or understood and that mental models were an important explanatory construct. In their analysis, diagrams are useful in concert with texts precisely because they assist the construction of mental models. This idea has been pursued in a very active program of work on multimedia instruction by Mayer and colleagues, which will be reviewed in the next section.

What about when the multiple sources of information are not presented as part of a single text, but rather independently, covering overlapping ground, so that the reader has to perform all the integration and mapping? This is the issue of multiple texts, and it has become commonplace in the age of the Internet. It is now rarely the case that a student struggles to find relevant source documents on a topic. Instead, students are typically faced with an overabundance of relevant materials and must somehow allocate their time across them, and integrate the knowledge they derive from different sources.

Perfetti (1997) has suggested that learning from multiple texts is one of the most important new challenges for text researchers. Research has shown, for example, that integrating information across multiple texts is a skill that does not come readily but can be acquired and taught (Stahl, Hind, Britton, McNish, & Bosquet, 1996; Rouet, Favart, Britt, & Perfetti, 1997).

The YSS theory raises important issues here. As previously noted, everyday reading of text can be seen as engendering a progression of mental representations moving from the surface form through the propositional content to a situation model. When reading, earlier representations can be discarded as later ones are formed, but for other tasks of text use, the reader needs to maintain a representation of the form of the multitext, and map this form onto the content. Payne and Reader (in press) refer to such a representation as a structure map.

The usefulness of a structure map becomes even more apparent when multiple texts are considered. In this case, structure maps could play a role in encoding *source* information, which might be important not only for locating information, but also for integrating diverse and potentially contradictory information and for making judgments of trust or confidence in the information. Source information might additionally encode temporal properties of information sources, and thus be useful for memory updating—revising knowledge in the light of new information, making distinctions between current and superseded propositions.

The widespread availability of the Web not only means that multiple texts are more widely encountered, but also encourages a situation where multiple texts are read in an interleaved fashion, in a single sitting, or at least temporally close, raising the importance of the above challenges, and meaning that recency in autobiographical memory is unlikely to accomplish source identification, so further stressing the importance of a structure map.

Payne and Reader (in press) studied readers' ability to search for specific ideas in multiple texts that they had just read. They found evidence that readers spontaneously constructed structure maps, as just described, in that they showed some memory of which documents contained which ideas, even when they did not expect to need such knowledge when reading the texts.

INSTRUCTIONS FOR MENTAL MODELS

Multimedia Instruction

If mental models are important for operating devices, how should they best be taught? We have seen that models are constructed automatically by readers of text, but can modern computational media, such as animations, be used to improve the acquisition of mental models from instructional texts, just as Glenberg and Langston (1992) suggested in the case of simple diagrams? Just such a question has been addressed in a longstanding programme of work by Richard Mayer and colleagues, which will be reviewed in this section.

Mayer and Moreno (2002) present a cognitive theory of multimedia learning, which builds on three main ideas:

1. From dual coding theory the authors suppose that humans have separate visual and verbal information processing systems (Clark & Paivio, 1991; Paivio, 1986);
2. From cognitive load theory the authors assume that the processing capacity of both the visual and the verbal memory system is strictly limited (Baddeley, 1992; Chandler & Sweller, 1991) and that cognitive load during instruction can interfere with learning;
3. From constructivist learning theory the authors take the idea that meaningful learning requires learners actively to select relevant information, to structure it into coherent representations, and make connections with other relevant knowledge (Mayer, 1996; Mayer, 1999a).

This latter process, of building coherent representations that connect information from different modalities with pre-existing knowledge bears clear relation to Johnson-Laird's construct of mental model, and indeed Mayer and colleagues use the term in this context. In the case of the physical systems that many of their studies have addressed, mental models may take the form of cause-effect chains. According to Mayer and Moreno (2002) a key design principle for instructional materials is that they should maximise the opportunity for these model-construction processes to be completed.

Mayer and colleagues have conducted a large number of experiments comparing learning from multimedia source materials with learning from components of these materials (words, pictures, etc) successively or in other kinds of combination. Based on this research, Mayer (1999b) and Mayer and Moreno (2002) have identified some principles of instructional design that foster multimedia learning.

The *multiple presentation principle* states that explanations in words and pictures will be more effective than explanations that use only words. (Mayer & Moreno, 2002, p. 107). When words only are presented, learners may find it difficult to construct an appropriate mental image, and this difficulty may block effective learning. Mayer and Anderson (1991; Experiment 2b) compared four treatment groups: words with pictures, words only, pictures only, and control, on tests of creative problem solving involving reasoning how a bicycle pump works. Results

demonstrated that participants in the words with pictures group generated a greater number of creative problem solutions than did participants in the other groups. Interestingly, animation without narration was equivalent to no instruction at all. Other studies have offered support for the general idea that learners will acquire richer knowledge from narration and animation than from narration alone (Mayer & Anderson, 1991, Experiment 2a; Mayer & Anderson, 1992, Experiments 1 and 2).

The *contiguity principle* is the claim that simultaneous as opposed to successive presentation of visual and verbal materials is preferred (Mayer & Moreno, 2002), because this will enable learners to build referential connections more readily (Mayer & Sims, 1994). Mayer and Anderson (1991, Experiments 1 and 2) studied a computer-based animation of how a bicycle pump works. They compared a version that presented words with pictures against the same content presenting words before pictures, and tested acquisition with tests of creative problem solving. Those in the words-with-pictures group generated about 50% more solutions to the test problems than did subjects in the words-before-pictures group.

The *individual differences principle* predicts that factors such as prior knowledge or spatial ability will influence transfer of learning from multimedia materials, moderating the effects of other principles (Mayer, 1999). With regard to domain specific knowledge, Mayer proposed that experienced learners may suffer little decrease in problem solving transfer when receiving narration and animation successively because their background knowledge will allow a mental model to be constructed from the words alone, then linked to the visual information. Low-experience learners on the other hand will have no means to over-ride the effects underlying the contiguity principle, and their problem solving transfer will suffer (Mayer & Sims, 1994). In support of this suggestion, experimental work by Mayer and Gallini (1990) demonstrated across three studies that the synchronization of words and pictures served to improve transfer for low- but not high-experience learners.

The *chunking principle* refers to a situation in which visual and verbal information must be presented successively, or alternately (against the contiguity principle). It states that learners will demonstrate better learning when such alternation takes place in short rather than long segments. The reasoning is straightforward, given the assumptions of the framework: working memory may become overloaded by having to hold large chunks before connections can be formed (Mayer, 1999b). An experiment by Mayer and Moreno (1998) investigated this *chunking principle* using explanations of how lightning storms develop. The ability to solve novel, transfer problems about lightning exhibited by a 'large chunk' group (who received all the visual information before or after all the verbal information) was compared with that of a 'small chunk' group (alternating presentations of a short portion of visual followed by a short portion of narration). The gain in performance of the small chunk group over the large chunk group was circa 100% (Mayer & Moreno, 1998).

The debt of Mayer's work to Sweller's programme of research on Cognitive Load Theory is obvious. Mayer's design principles reflect the premise that students will learn more deeply when their visual and/or verbal memories are not overloaded. Students are better able to make sense of information when they receive both verbal and visual representations rather than only verbal; when they can hold relevant visual and verbal representations in working memory at the same time; when they have domain specific knowledge and/or high spatial ability; and when they receive small bits of information at a time from each mode of presentation.

Despite incredibly positive research results, at this stage Mayer's work should be viewed with a little caution. Almost all of the experiments utilise very short instructional presentations, with some of the animations lasting only 30 seconds. Subjects are then required to answer problem-solving questions that seem ambiguous, requiring students to be fairly creative in order to generate solutions. Mayer's work also typically neglects to include any tests of long-term retention. It may conceivably be falling into the instructional trap of maximising performance during learning at the expense of longer-term performance. This issue is the focus of the next section.

The Theory of Learning by Not Doing

Mayer's theory of multimedia instruction adheres to the common assumption that the optimal design of instructional material involves minimizing the cognitive burden on the learner due to the limits of the working memory.

Yet minimizing the mental effort of learners is not necessarily or always a good instructional strategy. According to Schmidt and Bjork (1992), instructional conditions that achieve the training goals of generalizability and long-term retention are not necessarily those that maximize performance during the acquisition phase.

They argue that the goal of instruction and training in real-world settings should first be to support a level of performance in the long term, and second to support the capability to transfer that training to novel-tasks environments. Methodologically, in order to measure a genuine *learning effect,* some form of long-term assessment of retention must take place; skill acquisition is not a reliable indicator of learning.

Schmidt and Bjork (1992) discussed three situations in which introducing difficulties for the learner can enhance long-term learning. First, studies that vary the scheduling of tasks during practice were reported. Random practice is more difficult than blocked schedules of practice, as a given task is never practiced on the successive trial. Using a complex motor task involving picking up a tennis ball and using it to knock over a particular set of barriers, Shea and Morgan (1979) reported a clear advantage for subjects who practiced under blocked conditions (subsets of barriers to knock), in terms of performance during practice. However, the amount of learning as demonstrated by the retention phase favored the random condition. Similar results have been reported by Baddeley and Longman (1978), Lee and Magill (1983), and (with verbal tasks) Landauer and Bjork, 1978.

Schmidt and Bjork offer an explanation for this paradigm, in which retrieval practice may play a key role. They suggest that there may be a benefit, in terms of long-term retention, for activities that actually cause forgetting of the information to be recalled, forcing the learner to practice retrieving this information (Bjork & Allen, 1970).

Experiments that vary the feedback the learner receives have demonstrated a similar phenomenon. A study by Schmidt, Young, Swinnen, and Shapiro (1989) demonstrated that delaying the feedback that subjects received during motor tasks interfered with performance. However, on a delayed-retention test, those who had received the feedback least often demonstrated the most effective performance. This seems to contradict the established opinion that feedback is vital for effective learning. Schmidt and Bjork (1992) suggested that frequent feedback may actually serve to block information-processing activities that are important during the skill-acquisition phase.

A final area reviewed by Schmidt and Bjork concerns the introduction of variability during practice, such as when practicing tossing a beanbag at a target at a particular distance. Practicing at variable distances is more effective than practicing at a fixed distance (Kerr & Booth, 1978).

Does the Schmidt and Bjork approach extend to HCI tasks, and in particular to instruction for mental models?

One impressive example of an instructional effect in the Schimdt and Bjork (1992) paradigm is informed by the idea of mental models or situation models derived from text, as discussed in part 2 of this chapter. Informed by the distinction between a text base and a situation model, work by McNamara, Kintsch, Songer, and Kintsch (1996) has shown how expository text can be designed to introduce difficulties for readers in exactly the productive manner advocated by the Schmidt and Bjork conception of training. These authors created two versions of target texts, one more coherent than the other (one experiment used a text about traits of mammals, a second used a text about heart disease). Coherence cues were provided by linking clauses with appropriate connectives and by inserting topic headings. The level of readers' background knowledge on the topic of the text was also assessed with a pretest. After reading a text, participants were given tests of the text base (free recall of the text propositions and specific factual questions about the contents of the text) and tests of the situation model (problem-solving-based questions, questions requiring inferences from the text, and a concept-sorting task).

McNamara et al. (1996) reported that for measures that tested the text base, the high coherence texts produced better performance. However, for situation-model measures, test performance for high-knowledge readers was better when they read the low-coherence text. McNamara et al. argued that limiting the coherence of a text forced readers to engage in compensatory processing to infer unstated relations in the text. This compensatory processing supported a deeper understanding of the text, in that the information in the text became more integrated with background knowledge. Thus, for high-knowledge readers, the texts that were more difficult to read improved the situation model by encouraging more transfer-appropriate processing. Low-knowledge readers were, presumably, unable to achieve the compensatory inferences, and therefore did better with more coherent texts. Because the text base does not incorporate background knowledge, it was not enhanced by any compensatory processing. (This finding is related to the work of Mayer and Sims (1994) reviewed above.)

One very successful practical approach to the design of instructions for interactive devices which is well known in the HCI community, is perhaps quite strongly related to this more theoretically oriented work. The concept of a "minimal manual" was outlined by Carroll (1990). It sought to minimize the extent to which instructional materials obstruct learning. Crucially, a well-designed Minimal Manual does not necessarily optimize the speed at which users can perform procedures as they read. Carroll's manuals avoided explicit descriptions that encouraged rapid but mindless rote performance. Instead, the emphasis was on active learning whereby learners were encouraged to generate their own solutions to meaningful tasks. This process was facilitated in part by reducing the amount of text provided and including information about error recovery.

O'Hara and Payne (1998, 1999) argued that learning from a problem-solving experience might be enhanced to the extent that problem solvers planned their moves through the problem space. Many puzzles with an interactive user interface, and indeed many user interfaces to commercial systems, encourage a one-step-at-a-time approach to problem solving, in which a move is chosen from the currently available set. This may be quick and relatively effortless, yet lead to little learning and inefficient solutions. For example, in an HCI task, participants had to copy names and addresses from a database to a set of letters. Each item category from the database had to be copied to several letters, so that the most obvious and perhaps least effortful strategy of preparing letters one at a time was inefficient in terms of database access. O'Hara and Payne's manipulation was to increase the cost of making each move (in the copying experiment by adding a system lock-out time). This resulted in more planning, more think-time per move, meaning slower solutions in the first instance, but more efficient behavior in the longeterm, and the discovery of strategies that required fewer database accesses and fewer user inputs.

Recent work by Duggan and Payne (2001) combined several of the insights in the work just reviewed to explore acquisition of interactive procedures during instruction following. Good procedural instructions for interactive devices must satisfy two criteria. First, they must support performance. Like all procedural instructions they should effectively communicate the procedure they describe, so as to allow users who don't know the procedure to enact it successfully and efficiently. Second, they must support learning. In common with instructions for all procedures that will be used repeatedly, they should facilitate subsequent memory for the procedure, so that it might later be performed without consulting the instructions.

How might procedural instructions be designed so as to follow the Schmidt and Bjork paradigm and provide transfer-appropriate practice opportunities for the learner? Of course, not all manipulations that introduce difficulties during learning are beneficial for the learner. Simply making the instructions unclear is unlikely to be effective. However, much this idea may have informed the design of some commercial user manuals. The criterion that quality instructions must communicate the procedure that they describe cannot be ignored.

The work of Diehl and Mills (1995) further illustrated the relevance of the theory of text comprehension to the design of instruction for interactive procedures. They argued that in the case of procedural instructions the distinction between situation model and text base maps directly onto a distinction between memory for the procedure (as tested by later task performance) and memory for the instructions themselves.

Texts describing how to complete a task using a device (setting an alarm clock or constructing a child's toy) were provided. While reading a text, participants were required to either perform the task (read and do), or do nothing (read only). (In addition, Diehl and Mills studied some intermediate conditions, such as read and watch experimenter. These conditions produced intermediate results and are not relevant to the current argument.) The effect of these training methods was then examined by asking participants to recall the text and then complete the task.

Diehl and Mills reported that the increased exposure to the device in the read-and-do condition resulted in improved task performance times relative to the read-only condition. However, text recall was better in the read-only condition, supporting the conceptual separation of text base and situation model.

Inspired by this work, Duggan and Payne (2001) introduced a particular technique to exploit the principle of Schmidt and Bjork (1992) and the methods of McNamara and colleagues (1996). Like the manipulations of Diehl and Mills (1995), their innovation centered not on the design of the instructions per se, but rather on the way the instructions are read and used. Diehl and Mills' reported advantage for reading and doing over reading alone has no real practical implication, as it is difficult to imagine anyone advocating isolated reading as a preferred method. However, Duggan and Payne suggested that the way learners manage the interleaving of reading and doing will affect their later retention, and thus offers an important lever for improving instruction.

Many procedural instructions have a natural step-wise structure, and in these cases it is possible to execute the procedure while reading with minimal load on memory. Learners can read a single step, and then execute it before reading the next step. Such an approach is low in effort (and therefore attractive to the learner), but also low in transfer-appropriate practice and therefore, one would argue on the basis of the reviewed work, poor at encouraging retention. If learners could instead be prompted to read several procedural steps before enacting them, performance would be made more effortful, but learning might benefit. Readers would be encouraged to integrate the information across the chunk of procedural steps, and the increased memory load would provide transfer-appropriate practice.

Duggan and Payne (2001) developed this idea as follows. First, by implementing an online help system in the context of experimental tasks (programming a VCR) they forced participants into either a step-wise or a chunk-based strategy for interleaving reading and acting. These experiments demonstrated that reading by chunks did tax performance during training, but improved learning, in particular retention of the procedure. Next, they developed a more subtle, indirect manipulation of chunking. By adding a simple cost to the access of online instructions (c.f., O'Hara & Payne, 1998), they encouraged readers to chunk steps so as to minimize the number of times the instructions were accessed. Just as with enforced chunking, this led to improved retention of the procedures.

SHARED MENTAL MODELS

In the last 10 years or so there has been a rapid surge of interest in the concept of shared mental models in the domain of teamwork and collaboration. The use of mental models in this literature, to date, is somewhat inexact, with little theoretical force, except to denote a concern with what the team members know, believe, and want. As the name suggests, shared mental models refers to the overlap in individuals' knowledge and beliefs.

The central thesis and motive force of the literature is that team performance will improve when team members share relevant knowledge and beliefs about their situation, task, equipment, and team. Different investigations and different authors have stressed different aspects of knowledge, and indeed proposed different partitions into knowledge domains. (And recently, as we shall see, some investigators have questioned the extent to which overlapping knowledge is a good thing. There are some situations in which explicit distribution or division of knowledge may serve the team goals better.)

At first glance, the idea that teams need to agree about or share important knowledge seems intuitively plain. Models of communication i.e., Clark, 1992) stress the construction of a common ground of assumptions about each partner's background and intentions. The idea of shared mental models develops this idea in a plausible practical direction.

A recent study by Mathieu, Heffner, Goodwin, Salas, and Cannon-Bowers (2000) was one of the most compelling demonstrations of the basic phenomenon under investigation, as well as being centered on an HCI paradigm. For these reasons, this study will be described and used as a framework to introduce the space of theoretical and empirical choices that characterize the mainstream of the shared mental models literature.

Mathieu, Heffner, Goodwin, Salas, and Cannon-Bowers (2000) considered team members' mental models as comprising knowledge of four separate domains: (a) technology (essentially the mental models described in part one of this chapter); (b) job or task; (c) team interaction (such as roles, communication channels and information flow) and (d) other teammates' knowledge and attitudes. Knowledge of the last three types would rarely be called a mental model outside this literature, and so straight away we can see a broader and more practical orientation than in individually oriented mental models literatures.

For the purposes of operationalization, the authors suggested that these four categories of knowledge may be treated as two: task related and team related. This binary distinction mirrors a distinction that has been made in terms of team behaviors and communications, which have been considered in terms of a task track and a teamwork track (McIntyre & Salas, 1995).

Mathieu and colleagues studied dyads using a PC-based flight simulator. One member of each dyad was assigned to the joystick to control aircraft position. The other was assigned to keyboard, speed, weapon systems, and information gathering. Both members could fire weapons. The experimental procedure incorporated a training phase, including the task and basics of teamwork, and then the flying of six missions, divided into three equally difficult blocks of two, each mission lasting around 10 minutes. Performance on a mission was scored in terms of survival, route following, and shooting enemy planes. Team processes were scored by two independent raters viewing videotapes to assign scores, for, example, how well the dyad communicated with each other.

Mental models were measured after each pair of missions. At each measurement point, each individual's task or team

mental model was elicited by the completion of a relatedness matrix (one for task, one for team), in which the team member rated the degree to which each pair from a set of dimensions was related. For the task model there were eight dimensions, including diving versus climbing; banking or turning; and choosing airspeed. For the team model there were seven dimensions, including amount of information and roles and team spirit.

Thus, at each measurement point, participants had to assign numbers between −4 (negatively related, a high degree of one requires a low degree of the other) and +4 (positively related, a high degree of one requires a high degree of the other) to each pair of dimensions in each domain. For example, they had to rate the relatedness of diving versus climbing to choosing airspeed, and the relatedness of roles to team spirit. For each team at each time for each model-type a convergence index was calculated by computing a correlation co-efficient (QAP correlation) between the two matrices. The co-efficient could vary from −1 (complete disagreement) to +1 (completely shared mental models).

The main findings of this investigation were as follows. Contrary to hypothesis, convergence of mental models did not increase over time; rather it was stable across missions 1 to 3. This runs counter to a major and plausible assumption of the shared mental models program, which is that agreement between team members' should increase with extent of communication and collaboration.

Nevertheless, convergence of both task and team models predicted the quality of team process and the quality of performance. Further, the relationship between convergence and performance was fully mediated by quality of team process.

The most natural interpretation of these findings is that team process is supported by shared mental models. In turn, good team processes lead to good performance. According to its authors, this study provided the first clear empirical support for the oft-supposed positive relationship between shared mental models and team effectiveness (Mathieu et al., 2000, p. 280).

As well as being paradigmatic in illustrating the key ideas in the shared mental models literature, this study has several aspects that highlight the range of approaches and the controversy in the field.

First, it is worth considering what particular properties of the task and teams may have contributed to the positive relation between shared mental models and team process and performance. Compared with most situations in which coordination and collaboration are of prime interest, including most situations addressed by CSCW researchers, the teams studied by Matheiu et al. were minimal (two members) and the tasks were very short term and relatively circumscribed. Beyond these obvious remarks, I would add that the division of labor in the task was very "close," and the workers' performance was extremely interdependent. Of course, interdependence is the signature of collaborative tasks; nevertheless a situation in which one person controls airspeed and another controls altitude may make this interdependence more immediate than is the norm.

It is also possible that the relatively circumscribed nature of the task and collaboration contributed to the failure of this study to find evidence for the sharing of mental models increasing across the duration of collaboration.

As just mentioned, although the literature contains many proposals that shared mental models will positively influence process and performance, there has been much less empirical evidence. Another study of particular relevance to HCI is concerned with the workings of software development teams.

Software development is an ideal scenario for the study of team coordination for several reasons. First, much modern software development is quintessentially team based (Crowston & Kammerer, 1998; Curtis, Krasner, & Iscoe, 1998; Kraut & Streeter, 1995), and relies heavily on the complex coordinations of team members. Secondly, this effort is often geographically dispersed, further stressing collaboration and putting an emphasis on communications technologies. Finally, software development takes place in technologically advanced settings with technologically savvy participants, so that it provides something of a test bed for collaboration and communication technologies.

One study of complex geographically distributed software teams has been reported that partially supports the findings of the Mathieu et al. (2000) study and provided complementary evidence for positive effects of shared mental models on team performance. Espinosa, Kraut, Slaughter, Lerch, Herbsleb, and Mockus (2002) reported a multimethod investigation of software teams in two divisions of a multinational telecommunications company. The most relevant aspect of their study was a survey of 97 engineers engaged in team projects of various sizes ranging from 2 to 7. Team coordination and shared mental models (SMM) were both measured by simple survey items, followed by posthoc correlational analysis to uncover the relation between shared mental models and team process. As in the Mathieu et al. (2000) study, shared mental models were considered in two categories: task and team. A positive relation between team SMM and coordination was discovered, but the effect of task SMM was not significant.

It is worth being clear about the positive relation and how it was computed. Team SMM was computed for each team by assessing the correlations between each team member's responses to the team SMM survey items. This index was entered as an independent variable in a multiple regression to predict average reported levels of team coordination. It is, of course, hard to infer any causal relation from such correlational analyses, and one might also wonder about the validity of purely questionnaire-based measures of some of the constructs, yet nevertheless the study is highly suggestive that SMM can have a positive influence in group-work situations far removed from pairs of students interactive with a flight simulator. Additionally, Espinosa, Kraut, Slaughter, Lerch, Herbsleb, and Mockus (2002) reported an interview study in which respondents confirmed their own belief that SMM contributed positively to project communications and outcomes.

Nevertheless, Espinosa et al. (2002) failed to find any relation between task SMM and team process. It seems to me that, in view of the survey methodology, this would have been the more compelling evidence in favor of the SMM construct. It seems less surprising and perhaps less interesting that there should be a correlation between participants' survey responses concerning how well they communicated on their team, and, for example, their agreement about which teammates had high knowledge about the project.

Levesque, Wilson, and Wholey (2001) reported a different study of software development teams, using ad hoc student-project groupings to study whether sharing of Team SMM increased over time. They only measured Team SMM, using Likert scale items on which participants signaled amount of agreement or disagreement with statements like, "Most of our team's communication is about technical issues," "Voicing disagreement on this team is risky," or "Lines of authority on this team are clear.". Team SMM was measured by computing correlations among team members of these responses after 1, 2, and 3 months of working on a joint project.

Levesque, Wilson, and Wholey (2001) found that, contrary to their hypothesis, team SMM decreased over time. They argue that this is because projects were managed by a division of labor thath required much initial collaboration but meant that later activity was more individual.

There are surely many teamwork situations in which role differentiation is critical for success, and this observation suggested that the most straightforward interpretation of shared mental models is overly simple. Indeed, even in teams that continue to meet, communicate, and collaborate, it may be that role differentiation means that task mental models should not so much be "shared" as "distributed" to allow for effective team performance. (Studies of intimate couples have explored a similar process of specialization of memory functions, under the name "transactive memory," i.e., Wegner, 1987, 1995).

When roles are differentiated, it is no longer important that task knowledge is shared, but rather that individuals' knowledge about who knows what is accurate. Thus, one would expect team SMMs to support communication and collaboration even in teams with highly differentiated roles. This may explain the findings reviewed above. In the Mathieu et al. study, the team members' technical roles remained tightly interdependent, so that both task and team models had to be shared for successful performance. In the Espinosa et al. (2002) study, the technical roles may have been differentiated but the level of communication remained high, so that team SMM affected performance but task SMM did not. In the Levesque et al. study, the teams divided their labor to the extent that communication and collaboration ceased to be necessary (apart, perhaps for some final pooling of results). In this case, we would predict that neither task nor team SMMs would affect performance once the division of labor had been accomplished. No data on performance were reported, but team models became less shared over the course of the projects.

Although there has been quite a sudden flurry of interest in shared mental models, this brief review makes clear that much empirical and conceptual work remains to be done. Of particular relevance to this chapter is the question of what exactly is meant by a mental model in this context.

To date throughout the field, mental models have been considered as semantic knowledge, using traditional associative networks as a representation. Thus, mental models have typically been tapped using simple likert scales or direct questions about the relations (i.e. similarity) between constructs, analyzed with multidimensional techniques such as pathfinder (for a review of measurement techniques in this field, see Mohammed, Kilmoski, & Rentsch, 2000). Because interest has focused on the extent to which knowledge and beliefs are common among team members, these approaches have been useful, allowing quantitative measures of similarity and difference. Nevertheless, compared with the literature on individual mental models, they tend to reduce participants' understanding to mere associations, and yet the thrust of the individual work shows that this may not be appropriate, because the particular conceptualizations of the domain, the analogies drawn, the computational as well as informational relations between internal and external representations etc., can have real effects on performance. It seems that an important path of development may be to adopt this more refined cognitive orientation and investigate the impact of shared models—as opposed to shared networks of facts and associations—on collaboration.

References

Baddeley, A. (1992). Working memory. *Science*, *255,* 556–559.

Baddeley, A. D., & Longman, D. J. A. (1978). The influence of length and frequency of training session on the rate of learning to type. *Ergonomics, 21*, 627–635.

Bibby, P. A., & Payne, S. J. (1993). Internalization and use–specificity of device knowledge. *Human-Computer Interaction, 8,* 25–56.

Bibby, P. A., & Payne, S. J. (1996). Instruction and practice in learning about a device. *Cognitive Science, 20,* 539–578.

Bjork, R. A., & Allen, T. W. (1970). The spacing effect: Consolidation or differential encoding? *Journal of Verbal Learning and Verbal Behavior, 9,* 567–572.

Bransford, J. D., Barclay, J. R., & Franks, J. J. (1972). Sentence memory: a constructive versus interpretive approach. *Cognitive Psychology, 3,* 193–209.

Carroll, J. M. (1990). *The Nurnberg funnel: Designing minimalist instruction for practical computer skill,* Cambridge, MA: MIT Press.

Carroll, J. M., & Olson, J. R. (1988). Mental models in human-computer interaction. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 45–65). New York: Elsevier.

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction, 8,* 293–332.

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review, 3*(3), 149–170.

Clark, H. H. (1992). *Arenas of language use*. Chicago: Chicago University Press.

Cockburn, A., & Jones, S. (1996).Which way now? Analysing and easing inadequacies in WWW navigation. *International Journal of Human-Computer Studies, 45,* 195–130.

Collins, A., & Gentner, D. (1987). How people construct mental models. In D. Holland, & N. Quinn (Eds.), *Cultural models in language and thought.* Cambridge, UK: Cambridge University Press.

Craik, K. J. W. (1943). *The nature of explanation.* Cambridge: Cambridge University Press.

Crowston, K., & Kammerer, E. E. (1998). Coordination and collective mind in software requirements development. *IBM Systems Journal, 37*(2), 227–245.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM 31*(11), 1268–1286.

Diehl, V. A., & Mills, C. B. (1995). The effects of interaction with the device described by procedural text on recall, true/false, and task performance. *Memory and Cognition, 23*(6), 675–688.

Duggan, G. B., & Payne, S. J. (2001). Interleaving reading and acting while following procedural instructions. *Journal of Experimental Psychology: Applied, 7*(4), 297–307.

Ericsson, K. A., Krampe, R. T., & Tesch-Romer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review, 100*(3), 363–406.

Espinosa, J. A., Kraut, R. E., Slaughter, S. A., Lerch, J. F., Herbsleb, J. D., & Mockus, A. (2002). Shared mental models, familiarity and coordination: A multi-method study of distributed software teams. *Proceedings of the 23rd International Conference in Information Systems (ICIS)*, Barcelona, Spain, 425–433.[??? page numbers???]

Fletcher, C. R., & Chrysler, S. T. (1990). Surface forms, textbases and situation models: recognition memory for three types of textual information. *Discourse Processes, 13*, 175–190.

Gentner, D., & Stevens, A. L. (1983). *Mental models*. Hillsdale, NJ: Erlbaum.

Glenberg, A. M., & Langston, W. E. (1992). Comprehension of illustrated text: pictures help to build mental models. *Journal of Memory and Language, 31*, 129–151.

Halasz, F. G., & Moran, T. P. (1983). Mental models and problem-solving in using a calculator. *Proceedings of CHI 83 Human Factors in Computing Systems*, New York: ACM.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction*. Cambridge, MA: MIT Press.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, UK: Cambridge University Press.

Johnson-Laird, P. N. (1989). Mental models. In M. I. Posner (Ed.), *Foundations of cognitive science*. Cambridge, MA: MIT Press.

Kerr, R., & Booth, B. (1978). Specific and varied practice of a motor skill. *Perceptual and Motor Skills, 46*, 395–401.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to use a device. *Cognitive Science, 8*, 255–273.

Kintsch, W. (1998). *Comprehension*. Cambridge, UK: Cambridge University Press.

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development, *Communications of the ACM, 38*(3), 69–81.

Landauer, T. K., & Bjork, R. A. (1978). Optimum rehearsal patterns and name learning. In M. M. Gnineberg, P. E. Morris, & R. N. Sykes (Eds.), *Practical aspects of memory* (pp. 625–6321). London: Academic Press.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65–100.

Lee, T. D., & Magill, R. A. (1983). The locus of contextual interference in motorskill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9*, 730–746.

Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). Cognitive divergence and shared mental models in software development project teams, *Journal of Organizational Behavior, 22*, 135–144.

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. *Journal of Applied Psychology, 85*, 273–283.

Mayer, R. E. (1996). Learning strategies for making sense out of expository text: The SOI model for guiding three cognitive processes in knowledge construction. *Educational Psychology Review, 8*, 357–371.

Mayer, R. E. (1999a). Research-based principles for the design of instructional messages: The case of multimedia explanations. *Document Design, 1*, 7–20.

Mayer, R. E. (1999b). Multimedia aids to problem solving transfer. *International Journal of Educational Research, 31*, 611–623.

Mayer, R. E. (1999). Multimedia aids to problem-solving transfer. *International Journal of Educational Research, 31*, 661–624.

Mayer, R. E., & Anderson, R. B. (1992). The instructive animation: Helping students build connections between words and pictures in multimedia learning. *Journal of Educational Psychology, 84*, 444–452.

Mayer, R. E., & Anderson, R. B. (1991). Animations need narrations: An experimental test of a dual-coding hypothesis. *Journal of Educational Psychology, 83*, 484–490.

Mayer, R. E., & Gallini, J. K. (1990). When is an illustration worth ten thousand words? *Journal of Educational Psychology, 82*, 715–726.

Mayer, R. E., & Moreno, R. (1998). A split-attention affect in multimedia learning: Evidence for dual processing systems in working memory. *Journal of Educational Psychology, 90*, 312–320.

Mayer, R. E., & Moreno, R. (2002). Aids to computer-based multimedia learning. *Learning and Instruction, 12*, 107–119.

Mayer, R. E., & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology, 86*, 389–401.

Mayhew, D. J. (1992). *Principles and guidelines in software user interface design*. Englewood Cliffs, NJ: Prentice Hall.

McCloskey, M. (1983). Naïve theories of motion. In D. Gentner, & A. L. Stevens (Eds.), *Mental models* (pp. 299–323). Hillsdale, NJ: Erlbaum.

McIntyre, R. M., & Salas, E. (1995). Measuring and managing for team performance: Emerging principles from complex environments. In R. A. Guzzo & E. Salas (Eds.), *Team effectiveness and decision making in organizations* (pp. 9–45). San Francisco: Jossey-Bass.

McNamara, D. S., Kintsch, E., Songer, N. B., & Kintsch, W. (1996). Are good texts always better? Text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction, 14*, 1–43.

Moheammet, S., & Dunville, B. C. (2001). Team mental models in a team knowledge framework: Expanding theory and measurement across disciplinary boundaries. *Journal of Organizational Behavior, 22*(2), 89.

Mohammed, S., Klimoski, R., & Rentsch, J. R. (2000). The measurement of team mental models: We have no shared schema. *Organizational Research Methods, 3*(2), 123–165.

Moray, N. (1999). Mental models in theory and practice. In D. Gopher, & A. Koriat (Eds.), *Attention and performance XVII* (pp. 223–258). Cambridge, MA: MIT Press.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 7–14) Hillsdale, NJ: Erlbaum.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

O'Hara, K. P., & Payne, S. J. (1998). The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology, 35*, 34–70.

O'Hara, K. P., & Payne, S. J. (1999). Planning and user interface: The effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies, 50*, 41–59.

Schmidt, R. A., & Bjork, R. A. (1992). New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. *Psychological Science, 3*, 207–217.

Paivio, A. (1986). *Mental representations.* New York: Oxford University Press.

Payne, S. J. (1991). A descriptive study of mental models. *Behavior and Information Technology*, *10*, 3–21.

Payne, S. J. (2003). Users' mental models of devices: The very ideas. In. J.M. Carroll (Ed.), *HCI models, theories and frameworks*: *Towards a multi-disciplinary science* (pp. 135–156). San Francisco: Morgan Kaufmann.

Payne, S. J., Howes, A., & Hill, E. (1992). Conceptual instructions derived from an analysis of device models. *International Journal of Human-Computer Interaction, 4*, 35–58.

Payne, S. J., & Reader, W. R. (in press). Constructing structure maps of multiple on-line texts. *International Journal of Human-Computer Studies*.

Payne, S. J., Squibb, H. R., & Howes, A. (1990). The nature of device models: The yoked state space hypothesis and some experiments with text editors. *Human-Computer Interaction, 5*, 415–444.

Perfetti, C. A. (1997). Sentences, individual differences, and multiple texts. Three issues in text comprehension. *Discourse Processes, 23*, 337–355.

Radvansky, G. A., Spieler, D. H., & Zacks, R. T. (1993). Mental model organization. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19*, 95–114.

Radvansky, G. A., & Zacks, R. T. (1991). Mental models and fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*, 940–953.

Rouet, J. F., Favart, M., Britt, M. A., & Perfetti, C. A. (1997). Studying and using multiple documents in history: Effects of discipline expertise. *Cognition and Instruction, 75*(1), 85–106.

Schmidt, R. A., Young, D. E., Swinnen, S., & Shapiro, D. C. (1989). Summary knowledge of results for skill acquisition: Support for the guidance hypothesis. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 352–359.

Shea, J. B., & Morgan, R. L. (1979). Contextual interference effects on the acquisition, retention, and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory, 5*, 179–187.

Simon, H. A. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics, 69*, 99–118.

Simon, H. A. (1978). On the forms of mental representation. In C. W. Savage (Ed.), *Minnesota studies in the philosophy of science* (pp. 3–18) (Vol. 9). Minneapolis: University of Minnesota Press.

Simon, H. A. (1992). What is an "explanation" of behavior? *Psychological Science, 3*, 150–161.

Stahl, S. A., Hind, C. R., Britton, B. K., McNish, M. M., & Bosquet, D. (1996). What happens when students read multiple source documents in history. *Reading Research Quarterly, 31*(4), 430–456.

Tognazzini, B. (1992). *Tog on interface*. Reading, MA: Addison-Wesley.

van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension*. New York: Academic Press.

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In I. B. Mullen, & G. R. Goethals (Eds.), *Theories of group behaviour* (pp. 185–208). New York: Springer-Verlag.

Wegner, D. M. (1995). A computer network model of human transactive memory. *Social Cognition, 13*, 319–339.

Young, R. M. (1983). Surrogates and mappings. Two kinds of conceptual models for interactive devices. In D. Gentner, & A. L. Stevens (Eds.), *Mental models* (pp. 35–52). Hillsdale, NJ: Erlbaum.

◆4◆

EMOTION IN HUMAN-COMPUTER INTERACTION

Scott Brave and Clifford Nass
Stanford University

Understanding Emotion **78**
Distinguishing Emotion from Related Constructs . . . **79**
Mood 79
Sentiment 80
Effects of Affect **80**
Attention 80
Memory 81
Performance 81
Assessment 81
Causes of Emotion **82**
Needs and Goals 82
Appraisal Theories 83
Contagion 83
Moods and Sentiments 83
Previous Emotional State 83
Causes of Mood **84**
Contagion 84
Color 84
Other Effects 84
Measuring Affect **84**
Neurological Responses 85
Autonomic Activity 85
Facial Expression 85
Voice 86
Self-Report Measures 86
Affect Recognition by Users 87
Open Questions **88**
1. With which emotion should HCI designers be most concerned? 88
2. When and how should interfaces attempt to directly address users' emotions and basic needs (vs. application-specific goals)? 88
3. How accurate must emotion recognition be to be useful as an interface technique? 88
4. When and how should users be informed that their affective states are being monitored and adapted to? 88
5. How does emotion play out in computer-mediated communication (CMC)? 88
Conclusion **88**
Acknowledgments **88**
References **89**

Emotion is a fundamental component of being human. Joy, hate, anger, and pride, among the plethora of other emotions, motivate action and add meaning and richness to virtually all human experience. Traditionally, human-computer interaction (HCI) has been viewed as the "ultimate" exception; users must discard their emotional selves to work efficiently and rationality with computers, the quintessentially unemotional artifact. Emotion seemed at best marginally relevant to HCI and at worst oxymoronic.

Recent research in psychology and technology suggests a different view of the relationship between humans, computers, and emotion. After a long period of dormancy and confusion, there has been an explosion of research on the psychology of emotion (Gross, 1999). Emotion is no longer seen as limited to the occasional outburst of fury when a computer crashes inexplicably, excitement when a video game character leaps past an obstacle, or frustration at an incomprehensible error message. It is now understood that a wide range of emotions plays a critical role in *every* computer-related, goal-directed activity, from developing a three-dimensional (3D) CAD model and running calculations on a spreadsheet, to searching the Web and sending an e-mail, to making an online purchase and playing solitaire. Indeed, many psychologists now argue that it is impossible for a person to have a thought or perform an action without engaging, at least unconsciously, his or her emotional systems (Picard, 1997b).

The literature on emotions and computers has also grown dramatically in the past few years, driven primarily by advances in technology. Inexpensive and effective technologies that enable computers to assess the physiological correlates of emotion, combined with dramatic improvements in the speed and quality of signal processing, now allow even personal computers to make judgments about the user's emotional state in real time (Picard, 1997a). Multimodal interfaces that include voices, faces, and bodies can now manifest a much wider and more nuanced range of emotions than was possible in purely textual interfaces (Cassell, Sullivan, Prevost, & Churchill, 2000). Indeed, any interface that ignores a user's emotional state or fails to manifest the appropriate emotion can dramatically impede performance and risks being perceived as cold, socially inept, untrustworthy, and incompetent.

This chapter reviews the psychology and technology of emotion, with an eye toward identifying those discoveries and concepts that are most relevant to the design and assessment of interactive systems. The goal is to provide the reader with a more critical understanding of the role and influence of emotion, as well as the basic tools needed to create emotion-conscious and consciously emotional interface designs.

The "seat" of emotion is the brain; hence, we begin with a description of the psychophysiological systems that lie at the core of how emotion emerges from interaction with the environment. By understanding the fundamental basis of emotional responses, we can identify those emotions that are most readily manipulable and measurable. We then distinguish emotions from moods (longer-term affective states that bias users' responses to any interface) and other related constructs. The following section discusses the cognitive, behavioral, and attitudinal effects of emotion and mood, focusing on attention and memory, performance, and user assessments of the interface. Designing interfaces that elicit desired affective states requires knowledge of the causes of emotions and mood; we turn to that issue in the following section. Finally, we discuss methods for measuring affect, ranging from neurological correlates to questionnaires, and describe how these indicators can be used both to assess users and to manifest emotion in interfaces.

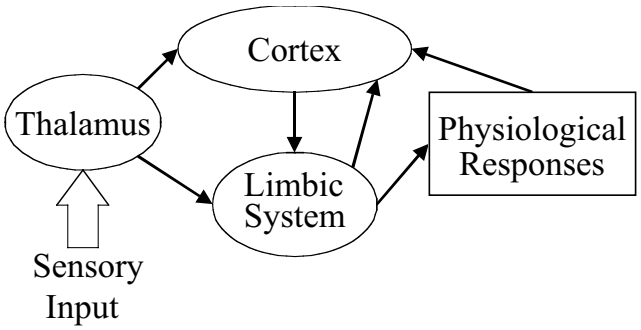


FIGURE 4.1. Neurological structure of emotion.

UNDERSTANDING EMOTION

What is emotion? Although the research literature offers a plethora of definitions (Kleinginna & Kleinginna, 1981), two generally agreed-upon aspects of emotion stand out: (a) emotion is a reaction to events deemed relevant to the needs, goals, or concerns of an individual; and, (b) emotion encompasses physiological, affective, behavioral, and cognitive components. Fear, for example, is a reaction to a situation that threatens (or seems to threaten, as in a frightening picture) an individual's physical well-being, resulting in a strong negative affective state, as well as physiological and cognitive preparation for action. Joy, on the other hand, is a reaction to goals being fulfilled and gives rise to a more positive, approach-oriented state.

A useful model for understanding emotion, based on a simplified view of LeDoux's (1996) work in neuropsychology, is shown in Fig. 4.1. There are three key regions of the brain in this model: (a) the thalamus, (b) the limbic system, and (c) the cortex. All sensory input from the external environment is first received by the thalamus, which functions as a basic signal processor. The thalamus then sends information simultaneously both to the cortex, for "higher-level" processing, and directly to the limbic system (LeDoux, 1995). The limbic system,[1] often called the "seat of emotion," constantly evaluates the need/goal relevance of its inputs. If relevance is determined, the limbic system sends appropriate signals both to the body, coordinating the physiological response, and also to the cortex, biasing attention and other cognitive processes.

[1]The limbic system is often considered to include the hypothalamus, the hippocampus, and the amygdala. According to LeDoux, the amygdala is the only critical area (LeDoux & Phelps, 2000).

The direct thalamic-limbic pathway is the mechanism that accounts for the more primitive emotions, such as startle-based fear, as well as innate aversions and attractions. Damasio (1994) called these the "primary" emotions. In an HCI context, on-screen objects and events have the potential to activate such primitive emotional responses (Reeves & Nass, 1996). For example, objects that appear or move unexpectedly (i.e. pop-up windows, sudden animations) and loud or sharp noises are likely to trigger startle-based fear. Visual stimuli that tend to be particularly arousing include images that fill a large fraction of the visual field (either because the image or screen is large or because the eyes are close to the screen; (Detenber & Reeves, 1996; Voelker, 1994)), images that seem to approach the user (i.e., a rapidly expanding image on the screen, an image that appears to be flying out from the screen, or a character that walks toward the user), and images that move in peripheral vision (i.e., on the side of the screen; (Reeves & Nass, 1996)). Finally, certain images and sounds may be innately disturbing or pleasing due to their evolutionary significance (i.e., screeching or crying noises or explicit sexual or violent imagery; (see, i.e., Lang, 1995; Malamuth, 1996)).

Most of the emotions that we are concerned with in the design of HCI—and the ones we will focus on in the remainder of this chapter—require more extensive cognitive (i.e., knowledge-based) processing. These "secondary" emotions, such as frustration, pride, and satisfaction, result from activation of the limbic system by processing in the cortex. Such cortical processing can occur at various levels of complexity, from simple object recognition (i.e., seeing the Microsoft Office Paperclip) to intricate rational deliberation (i.e., evaluating the consequences of erasing a seldom-used file), and may or may not be conscious. The cortex can even trigger emotion in reaction to internally generated stimuli (i.e., thinking about how difficult it will be to configure a newly purchased application).

Finally, an emotion can result from a combination of both the thalamic-limbic and the cortico-limbic mechanisms. For example, an event causing an initial startle/fear reaction can be later recognized as harmless by more extensive, rational evaluation (i.e., when you realize that the flash of your screen suddenly going blank is just the initiation of the screen saver). In other situations, higher-level processing can reinforce an initial evaluation. Whatever the activation mechanism—thalamic or cortical, conscious or nonconscious—the cortex receives input from an activated limbic system, as well as feedback from the body, both contributing to the conscious "experience" of emotion.

The previous discussion provides a useful framework for considering one of the classic debates in emotion theory: are emotions innate or learned? At one extreme, evolutionary theorists argue that all emotions (including complex emotions such as regret and relief) are innate, each evolved to address a specific environmental concern of our ancestors (Darwin, 1872/1998; Neese, 1990; Tooby & Cosmides, 1990; see also Ekman, 1994; Izard, 1992). These theories are consistent with a hypothesis of high differentiation within the limbic system, corresponding to each of the biologically determined emotions. From this perspective, it is also reasonable to speculate that each emotion is associated with a unique set of physiological and cognition-biasing responses.

At the other extreme, many emotion theorists argue that, with the exception of startle and innate affinity/disgust (which they would consider pre-emotional); emotions are almost entirely learned social constructions (Averill, 1980; Ortony & Turner, 1990; Shweder, 1994; Wierzbicka, 1992). Such theories emphasize the role of higher cortical processes in differentiating emotions and concede minimal, if any, specificity within the limbic system (and consequently, within physiological responses). For example, the limbic system may operate in simply an on/off manner, or at most be differentiated along the dimensions of valence (positive/negative or approach/avoidance) and arousal (low/high) (Barrett & Russell, 1999; Lang, 1995). From this perspective, emotions are likely to vary considerably across cultures, with any consistency being based in common social structure, not biology.

Between these two extremes lie those who believe that there are "basic emotions." Citing both cross-cultural universals and primate studies, these theorists contend that there is a small set of innate, basic emotions shared by all humans (Ekman, 1992; Oatley & Johnson-Laird, 1987; Panksepp, 1992). Which emotions qualify as basic is yet another debate, but the list typically includes fear, anger, sadness, joy, disgust, and sometimes also interest and surprise. Other emotions are seen either as combinations of these basic emotions or as socially learned differentiations within the basic categories (i.e., agony, grief, guilt, and loneliness are various constructions of sadness; (Bower, 1992). In this view, the limbic system is prewired to recognize the basic categories of emotion, but social learning and higher cortical processes still play a significant role in differentiation.

If the "basic emotions" view is correct, a number of implications for interaction design and evaluation emerge. First, the basic categories would likely be the most distinguishable, and therefore measurable, emotional states (both in emotion recognition systems as well as in postinteraction evaluations). Further, the basic emotions would be less likely to vary significantly from culture to culture; facilitating the accurate translation and generalizability of questionnaires intended to assess such emotions. Lower variability also enables more reliable prediction of emotional reactions to interface content, both across cultures and across individuals. Finally, for users interacting with on-screen characters, depictions of the basic emotions would presumably be most immediately recognizable. If the social construction view of emotions is valid, then emotion measurement and assessment, prediction, and depictions are more challenging and nuanced.

DISTINGUISHING EMOTION FROM RELATED CONSTRUCTS

Mood

It is useful to distinguish among several terms often used ambiguously: emotion, mood, and sentiment. Emotion can be distinguished from mood by its object-directedness. As Frijda (1994) explained, emotions are *intentional*: They "imply and involve relationships with a particular object." We get scared *of* something, angry *with* someone, and excited *about* some event.

Moods, on the other hand, though they may be indirectly caused by a particular object, are "nonintentional"; they are not *directed* at any object in particular and are thus experienced as more diffuse, global, and general. A person can be sad about something (an emotion) or generally depressed (a mood). Unfortunately, the English language often allows the same term to describe both emotion and mood (i.e., "happy").

Another distinction between emotion and mood emerges from a functional perspective. As a reaction to a particular situation, emotions bias action—they prepare the body and the mind for an appropriate, immediate response. As such, emotions also tend to be relatively short lived. Moods, in contrast, tend to bias cognitive strategies and processing over a longer term (Davidson, 1994). More generally, moods can be seen to serve as a background affective filter through which both internal and external events are appraised. A person in a good mood tends to view everything in a positive light, while a person in a bad mood does the opposite. The interaction between emotions and moods is also important. Moods tend to bias which emotions are experienced, lowering the activation thresholds for mood-related emotions. Emotions, on the other hand, often cause or contribute to moods.

When assessing user response to an interface, it is important to consider the biasing effects of user mood. Users entering a usability or experimental study in a good mood, for instance, are more likely to experience positive emotion during an interaction than users in a bad mood. Pretesting for mood and including it as a variable in analysis can, therefore, reduce noise and increase interpretive power. If pretesting users immediately prior to an interaction is inappropriate, there is a second noise-reducing option: assessment of temperament. Temperament reflects the tendency of certain individuals to exhibit particular moods with great frequency. Participants can be pretested for temperament at any point prior to the study, enabling the exclusion of extreme cases of depressive or excitable individuals (i.e., Bishop, Jacks, & Tandy, 1993). Finally, if user testing involves multiple stimuli, order of presentation can also influence the results. For example, earlier stimuli may establish a mood that biases emotional reactions to subsequent stimuli. To combat this problem, the order of stimuli should be varied from participant to participant, when feasible.

Sentiment

Sentiment is also often confused with emotion. Unlike emotions (and moods), sentiments are not states of an individual, but assigned properties of an object. When people say that they "like" an interface or find an interface to be "frustrating," what they really mean is that that they associate the interface with a positive or frustrating emotional state; in other words, they expect interaction with the interface to lead to positive or frustrating emotions. The basis for this judgment often comes from direct experience and subsequent generalization, but may also arise from social learning (Frijda, 1994).

One reason for the confusion between emotions and sentiment is that many languages use the same words for both. For example, the word "like" can be used both to indicate prediction or opinion (sentiment) as well as a current emotional state (i.e. "I like receiving e-mail" vs. "I like the e-mail that just arrived"). Clore (1994, p. 108) offers an interesting explanation for this ambiguity, theorizing that sentiments are judged by bringing the object to mind and observing the affective reaction. But, while emotions and moods are fleeting—emotions last only seconds and moods last for hours or even days—sentiments can persist indefinitely and are thus responsible for guiding our propensities to seek out or avoid particular objects and situations. In this sense, sentiments are of critical importance for HCI because they motivate users to return to particular software products or Websites.

Although direct interaction with an object is the most accurate way for a user to create a sentiment (consider the colloquial phrase, "how do you know you don't like it unless you try it"), sentiments can also be caused by assumptions based on the communicated properties of an object. People may, for example, base a sentiment on someone else's description of their interaction with the object, or even immediately adopt the sentiment of someone they know or respect (i.e., consider the presumed influence of celebrities in software advertisements).

As a predictive construct, sentiments are often generalizations about a class of objects with a given recognizable property, i.e., stereotypes. Although some of these generalizations may be logical and accurate, others may not—in fact, they may not even be conscious. Negative experiences with a particular computer character, for example, may lead users to conclude that they dislike all character-based interfaces. However, using a character that people know and like already—Mickey Mouse, for example—may be able to leverage sentiment to an interface's advantage. Similarly, many people have well-established sentiments regarding certain types of applications (i.e. "I hate spreadsheet applications"). For such users, interfaces that avoid triggering their negative stereotypes have the advantage. Positive stereotypes, on the other hand, should be encouraged whenever possible, such as when learning applications are framed as entertainment.

EFFECTS OF AFFECT

Attention

One of the most important effects of emotion lies in its ability to capture attention. Emotions have a way of being completely absorbing. Functionally, they direct and focus our attention on those objects and situations that have been appraised as important to our needs and goals so that we can deal with them appropriately. Emotion-relevant thoughts then tend to dominate conscious processing—the more important the situation, the higher the arousal, and the more forceful the focus (Clore & Gasper, 2000). In an HCI context, this attention-getting function can be used advantageously, as when a sudden beep is used to alert the user, or can be distracting, as when a struggling user is frustrated and can only think about his or her inability.

Emotion can further influence attention through a secondary process of emotion regulation (Gross, 1998). Once an

emotion is triggered, higher cognitive processes may determine that the emotion is undesirable. In such cases, attention is often directed away from the emotion-eliciting stimulus for the purpose of distraction. For example, becoming angry with an onscreen agent may be seen as ineffectual (i.e., because it doesn't recognize your anger) or simply unreasonable. An angered user may then actively try to ignore the agent, focusing instead on other onscreen or off-screen stimuli, or even take the next step and completely remove the agent from the interaction (which could mean leaving an application or Website entirely). Positive emotions may likewise require regulation at times, such as when amusing stimuli lead to inappropriate laughter in a work environment. If the emotionally relevant stimulus is too arousing, however, regulation through selective attention is bound to fail (Wegner, 1994), because users will be unable to ignore the stimulus.

Mood can have a less profound but more enduring effect on attention. At the most basic level, people tend to pay more attention to thoughts and stimuli that have some relevance to their current mood state (Bower & Forgas, 2000). However, people also often consciously regulate mood, selecting and attending to stimuli that sustain desired moods or, alternatively, counteract undesired moods. An interface capable of detecting—or at least predicting—a user's emotional or mood state could similarly assume an affect-regulation role, helping to guide attention away from negative and toward more positive stimuli. For example, a frustrated user could be encouraged to work on a different task, focus on a different aspect of the problem at hand, or simply take a break (perhaps by visiting a suggested online entertainment site).

Memory

Emotion's effect on attention also has implications for memory. Because emotion focuses thought on the evoking stimulus, emotional stimuli are generally remembered better than unemotional events (Thorson & Friestad, 1985). Negative events, which tend to be highly arousing, are typically remembered better than positive events (Newhagen & Reeves, 1991, 1992; Reeves & Nass, 1996, Chapter 10; Reeves, Newhagen, Maibach, Basil, & Kurz, 1991). In addition, emotionality "improves memory for central details while undermining memory for background details" (see Heuer & Reisberg, 1992; Parrott & Spackman, 2000).

Mood also comes into play in both memory encoding and retrieval. Research has shown that people will remember "mood-congruent" emotional stimuli better than incongruent stimuli. Bower, Gilligan, and Monteiro (1981), for example, hypnotized subjects into either a happy or sad mood before having them read stories about various characters. The next day, subjects were found to remember more facts about characters whose mood had agreed with their own than about other characters. Similarly, on the retrieval end, people tend to better recall memories consistent with their current mood (Ellis & Moore, 1999). However, the reverse effect has also been shown to occur in certain situations; people will sometimes better recall mood-incongruent memories (i.e., happy memories while in a sad mood). Parrott and Spackman (2000) hypothesized that mood regulation is responsible for this inverse effect: When a given mood is seen as inappropriate or distracting, people will often actively try to evoke memories or thoughts to modify that mood (see Forgas, 1995) Affect Infusion Model (AIM) for insight into these contradictory findings (also see Erber & Erber, 2001)). Finally, there is some evidence for mood-dependent recall: Memories encoded while in a particular mood are better recalled when in that same mood. This effect is independent of the emotional content of the memory itself (Ucros, 1989). It should be noted, however, that the effects of mood on memory are often unreliable and therefore remain controversial.

Performance

Mood has also been found to affect cognitive style and performance. The most striking finding is that even mildly positive affective states profoundly affect the flexibility and efficiency of thinking and problem solving (Hirt, Melton, McDonald, & Harackiewicz, 1996; Isen, 2000; Murray, Sujan, Hirt, & Sujan, 1990). In one of the best-known experiments, subjects were induced into a good or bad mood and then asked to solve Duncker's (1945) candle task. Given only a box of thumbtacks, the goal of this problem was to attach a lighted candle to the wall, such that no wax drips on the floor. The solution required the creative insight to thumbtack the box itself to the wall and then tack the candle to the box. Subjects who were first put into a good mood were significantly more successful at solving this problem (Isen, Daubman, & Nowicki, 1987). In another study, medical students were asked to diagnose patients based on X-rays after first being put into a positive, negative, or neutral mood. Subjects in the positive-affect condition reached the correct conclusion faster than did subjects in other conditions (Isen, Rosenzweig, & Young, 1991). Positive affect has also been shown to increase heuristic processing, such as reliance on scripts and stereotypes. Though some have argued that such reliance is at the expense of systematic processing (Schwartz & Bless, 1991), more recent evidence suggests that heuristic processing and systematic processing are not mutually exclusive (Isen, 2000). Keeping a user happy may, therefore, not only affect satisfaction, but may also lead to efficiency and creativity.

Assessment

Mood has also been shown to influence judgment and decision making. As mentioned earlier, mood tends to bias thoughts in a mood-consistent direction, while also lowering the thresholds of mood-consistent emotions. One important consequence of this is that stimuli—even those unrelated to the current affective state—are judged through the filter of mood (Clore et al., 2001; Erber & Erber, 2001; Niedenthal, Setterlund, & Jones, 1994). This suggests that users in a good mood will likely judge both the interface and their work more positively, regardless of any direct emotional effects. It also suggests that a happy user at an e-commerce site would be more likely to evaluate the products or services positively.

Positive mood also decreases risk-taking, likely in an effort to preserve the positive mood. That is, although people in a positive mood are more risk-prone when making hypothetical decisions, when presented with an actual risk situation, they tend to be more cautious (Isen, 2000). In an e-commerce purchasing situation, then, one can predict that a low-risk purchase is more likely during a good mood, due to a biased judgment in favor of the product, while a high-risk purchase may be more likely in a less cautious, neutral, or negative mood (consistent with the adage that desperate people resort to desperate measures).

A mood's effect on judgment, combined with its effect on memory, can also influence the formation of sentiments. Sentiments are not necessarily determined during interaction with an object; they often are grounded in reflection. This is important to consider when conducting user tests, as the mood set by the interaction immediately prior to a questionnaire may bias like/dislike assessments of earlier interactions. Thus, varying order of presentation ensures both that later stimuli do not influence the *assessment* of earlier stimuli, and that earlier stimuli do not influence the *experience* of later stimuli (as discussed earlier).

CAUSES OF EMOTION

What causes emotions? The answer to this question is critical for HCI because an understanding of emotions' antecedents will better enable us to design interfaces that encourage desired emotional states and understand interfaces that do not.

Needs and Goals

As we discussed in the first section, emotions are reactions to situations deemed relevant to the needs and goals of the individual. Clearly, a user comes to a computer hoping to achieve certain application-specific goals—composing a document, sending an e-mail, finding a piece of information, etc. The degree to which an interface facilitates or hampers those goals has a direct effect on the emotional state of the user. An interface capable of detecting emotion could, therefore, use such information as feedback regarding whether the user's goals are being met, modifying its behavior as necessary. In an information-seeking context, for example, emotional reactions to displayed content could be used to improve the goal-relevance of future retrievals. Similarly, if an interface detects frustration, desperation, or anger in a user, goals may be facilitated by trying a new approach or offering assistance (Klein, Moon, & Picard, 1999; Picard, 1997a). (If the particular goals implicated by an emotion are not clear, there can be advantages to an interface that empathizes with the user; (Klein et al., 1999)). More generally, user preferences can be automatically determined based on a user's emotional reactions to interface elements (Picard, 1997a).

There are also a host of more abstract needs underlying, and often adjacent to, application-specific goals. A user may have a strong need to feel capable and competent, maintain control, learn, or be entertained. A new user typically needs to feel comfortable and supported, while an expert is more focused on aesthetic concerns of efficiency and elegance. Acknowledging these more abstract goals in interface design can be as instrumental in determining a user's affective state as meeting or obstructing application-specific goals. Maslow's hierarchy (Maslow, 1968) presents a useful starting place for considering the structure of these more abstract user needs. In his later work, Maslow (1968) grouped an individual's basic needs into eight categories:

- **Physiological:** hunger, thirst, bodily comforts, etc.
- **Safety/security:** being out of danger
- **Social:** affiliate with others, be accepted
- **Esteem:** to achieve, be competent, gain approval and recognition
- **Cognitive:** to know, to understand, and explore
- **Aesthetic:** symmetry, order, and beauty
- **Self-actualization:** to find self-fulfillment and realize one's potential
- **Transcendence:** to help others find self-fulfillment and realize their potential.

When a particular situation or event is deemed as promoting these needs, positive emotion results. When someone or something hampers these needs, negative emotion results. The specific emotion experienced is due in part to the category of need implicated by the event. Fright, for example, is typically associated with threatened safety/security needs; love and embarrassment with social needs; pride with esteem needs; and curiosity with cognitive needs.

Within Maslow's (1968) framework, application-specific goals of a user can be seen as instruments ultimately serving these more basic needs. For example, a user who successfully enhances a digital family photograph may simultaneously be contributing to the fulfillment of social, esteem, cognitive, and aesthetic needs. However, interfaces can also *directly* address a user's basic needs. For example, a spell-checker interface that praises a user on his or her spelling ability, regardless of the user's actual performance, is a somewhat humorous, though illustrative, approach to acknowledging a user's esteem needs. Such interfaces, by enhancing the user's affective state, have been shown also to be viewed as more intelligent and likable (Reeves & Nass, 1996, Chapter 4). As another example, an interface that takes care to establish a trusting and safe relationship with users may ultimately lead to more effective and cooperative interactions (Fogg, 1998). Educational software should address users' emotional needs, not only teaching the relevant content, but also ensuring users *believe* that they are learning. Optimized learning further requires a careful balance of esteem and self-actualization needs, offering appropriate levels of encouragement and challenge, as well as praise and criticism. Finally, one of the key arguments for social interfaces is that they meet the social needs of users (Reeves & Nass, 1996).

Although the type of need relevant in a situation offers some insight into emotional reaction, need category alone is not sufficient to differentiate fully among all emotions. Distinguishing frustration and anger, for example, cannot be achieved based solely on knowing the users' need; it also requires some notion of agency.

Appraisal Theories

"Appraisal" theories provide much greater predictive power than category or hierarchy-based schemes by specifying the critical properties of antecedent events that lead to particular emotions (Lazarus, 1991; Ortony, Clore, & Collins, 1988; Roseman, Antoniou, & Jose, 1996; Scherer, 1988). Ellsworth (1994), for example, described a set of "abstract elicitors" of emotion. In addition to *novelty* and *valence*, Ellsworth contended that the level of *certainty/uncertainty* in an event has a significant impact on the emotion experienced. For instance, "uncertainty about probably positive events leads to interest and curiosity, or to hope," while, "uncertainty about probably negative events leads to anxiety and fear" (Ellsworth, 1994, p. 152). Certainty, on the other hand, can lead to relief in the positive case and despair in the negative case.

Because slow, unclear, or unusual responses from an interface generally reflect a problem, one of the most common interface design mistakes—from an affective standpoint—is to leave the user in a state of uncertainty. Users tend to fear the worst when, for example, an application is at a standstill, the hourglass remains up longer than usual, or the hard drive simply starts grinding away unexpectedly. Such uncertainty leads to a state of anxiety that can be easily avoided with a well-placed, informative message or state indicator. Providing users with immediate feedback on their actions reduces uncertainty, promoting a more positive affective state (see Norman, 1990, on visibility and feedback). When an error has *actually* occurred, the best approach is to make the user aware of the problem and its possible consequences, but frame the uncertainty in as positive a light as possible (i.e., "this application has experienced a problem, but the document should be recoverable").

According to Ellsworth (1994), *obstacles and control* also play an important role in eliciting emotion. High control can lead to a sense of challenge in positive situations, but stress in negative situations. Lack of control, on the other hand, often results in frustration, which if sustained can lead to desperation and resignation. In an HCI context, providing an appropriate level of controllability, given a user's abilities and the task at hand, is thus critical for avoiding negative affective consequences. Control need not only be perceived to exist (Skinner, 1995; Wegner, Bargh, Gilbert, Fiske, et al., 1998), but must be understandable and visible, otherwise the interface itself is an obstacle (Norman, 1990).

Agency is yet another crucial factor determining emotional response (Ellsworth, 1994; Friedman & Kahn, 1997). When oneself is the cause of the situation, shame (negative) and pride (positive) are likely emotions. When another person or entity is the cause, anger (negative) and love (positive) are more likely. However, if fate is the agent, one is more likely to experience sorrow (negative) and joy (positive). An interface often has the opportunity to direct a user's perception of agency. In any anomalous situation, for example—be it an error in reading a file, inability to recognize speech input, or simply a crash—if the user is put in a position encouraging blame of oneself or fate, the negative emotional repercussions may be more difficult to diffuse than if the computer explicitly assumes blame (and is apologetic). For example, a voice interface encountering a recognition error can say, "This system failed to understand your command" (blaming itself), "The command was not understood" (blaming no one), or "You did not speak clearly enough for your command to be understood" (blaming the user).

Appraisal theories of emotion, such as Ellsworth's (1994), are useful not only in understanding the potential affective impacts of design decisions, but also in creating computer agents that exhibit emotion. Although in some cases scripted emotional responses are sufficient, in more dynamic or interactive contexts, an agent's affective state must be simulated to be believable. Ortony, Clore, and Collins' (1988) cognitive theory of emotion is currently the most commonly applied appraisal theory for such purposes (Bates, Loyall, & Reilly, 1994; Elliott & Brzezinski, 1998; for alternate approaches, see Ball & Breese, 2000; Bozinovski & Bozinovska, 2001; Scheutz, Sloman, & Logan, 2000). Appraisal theories can also be used to help model and predict a user's emotional state in real time (Elliott & Brzezinski, 1998).

Contagion

Another cause of emotion that does not fit cleanly into the structure just described is contagion (Hatfield, Cacioppo, & Rapson, 1994). People often "catch" other's emotions. Sometimes this social phenomenon seems logical, such as when a person becomes afraid upon seeing another experience fear. At other times, contagion seems illogical, such as when another person's laughter induces immediate, "unexplainable" amusement. Anticipatory excitement is another emotion that transfers readily from person to person.

Emotions *in* interfaces can also be contagious. For example, a character that exhibits excitement when an online product appears can make users feel more excited. Similarly, an attempt at light humor in a textual interface, even if unsuccessful, may increase positive affect (Morkes, Kernal, & Nass, 2000).

Moods and Sentiments

Mood and sentiment can also bias emotion. One of the fundamental properties of mood is that it lowers the activation threshold for mood-consistent emotions. Sentiment can act in a similar way. For example, interaction with an object, to which a sentiment is already attached, can evoke emotion either in memory of past interaction or in anticipation of the current interaction. Thus, an interface that proved frustrating in the past may elicit frustration before the user even begins working. In addition, sentiment can bias perception of an object, increasing the probability of eliciting sentiment-consistent emotions. For example, an application that users *like* can do no wrong, while one that users *dislike* does everything to anger them, regardless of the application's actual behavior. Of critical importance here is that sentiments need not derive from direct experience; they may also be inferred from stereotypes or other generalizations.

Previous Emotional State

Finally, a user's previous emotional state can affect the experience of subsequent emotions. This occurs not only through the

mechanism of mood—emotions can cause moods and moods then bias the activation thresholds of emotions—but also through the mechanisms of excitation transfer and habituation. Excitation transfer (Zillmann, 1991) is based on the fact that after an emotion-causing stimulus has come and gone, an activated autonomic nervous system takes some time to return to its deactivated state. If another emotion is triggered before that decay is complete, the residual activation ("excitement") will be added to the current activation and be perceived as part of the current emotion. As Zillmann (1991) explained, "residues of excitation from a previous affective reaction will *combine* with excitation produced by subsequent affective stimulation and thereby cause an *overly intense* affective reaction to subsequent stimuli. . . . Residual arousal from anger, then, may intensify fear; residues from fear may intensify sexual behaviors; residual sexual arousal may intensify aggressive responses; and so forth" (p. 116). Thus, people who have just hit the "purchase" button associated with their web shopping cart can become particularly angry when they are presented with multiple pages before they can complete their transaction: The arousal of buying increases the intensity of their frustration with the post-purchase process. Similarly, Reeves and Nass (1996) have argued that pictorial characters "raise the volume knob" on both positive and negative feelings about an interaction, because explicitly social interactions are more arousing than their non-social counterparts are.

Habituation is, in some sense, the converse of excitation transfer. It posits that the intensity of an emotion *decreases* over time if the emotion is experienced repeatedly. One explanation for this effect relates back to appraisal theory: "Emotions are elicited not so much by the presence of favorable or unfavorable conditions, but by actual or expected changes in favorable or unfavorable conditions" (Frijda, 1988, p. 39). Repeated pleasurable affective states, therefore, become expected and thus gradually lose intensity. The same is true for negative affective states; however, particularly extreme negative emotional states may never habituate (Frijda, 1988). This may be why negative experiences with frequently used interfaces (i.e., operating systems) are remembered more vividly than positive experiences.

CAUSES OF MOOD

Mood has a number of potential causes. The most obvious is emotion itself. Intense or repetitive emotional experiences tend to prolong themselves into moods. A user who is continually frustrated will likely be put in a frustrated mood, while a user who is repeatedly made happy will likely be put in a positive mood. Mood can also be influenced, however, by anticipated emotion, based on sentiment. For example, if users know that they must interact with an application that they dislike (i.e., they associate with negative emotion), they may be in a bad mood from the start.

Contagion

Similar to emotion, moods also exhibit a contagion effect (Neumann & Strack, 2000). For example, a depressed person will often make others feel depressed and a happy person will often make others feel happy. Murphy and Zajonc (1993) have shown that even a mere smiling or frowning face, shown so quickly that the subject is not conscious of seeing the image, can affect a person's mood and subsequently bias judgment. From an interface standpoint, the implications for character-based agents are clear: Moods exhibited by onscreen characters may directly transfer to the user's mood. Onscreen mood can also lead to "perceived contagion" effects: One smiling or frowning face on the screen can influence users' perceptions of other faces that they subsequently see on the screen, perhaps because of priming (Reeves, Biocca, Pan, Oshagan, & Richards, 1989; Reeves & Nass, 1996, Chapter 22).

Color

Color can clearly be designed into an interface with its mood influencing properties in mind. Warm colors, for example, generally provoke "active feelings," while cool colors are "much less likely to cause extreme reactions" (Levy, 1984). Gerard (1957; 1958), for example, found that red light projected onto a diffusing screen produces increased arousal in subjects, using a number of physiological measures (including cortical activation, blood pressure, and respiration), while blue light has essentially the opposite "claming" effect (see Walters, Apter, & Svebak, 1982). Subjective ratings of the correlations between specific colors and moods can be more complicated. As Gardano (1986) summarized, "yellow (a warm color) has been found to be associated with both sadness (Peretti, 1974) and with cheerfulness (Wexner, 1954). Similarly, red (another warm color) is related to anger and violence (Schachtel, 1943) as well as to passionate love (Henry & Jacobs, 1978; Pecjak, 1970); and blue (a cool color), to tenderness (Schachtel, 1943) and sadness (Peretti, 1974). . . ." Nevertheless, as any artist will attest, carefully designed color schemes (combined with other design elements) can produce reliable and specific influences on mood.

Other Effects

A number of other factors can affect mood. For example, in music, minor scales are typically associated with negative emotion and mood, while major scales have more positive/happy connotations (Gregory, Worrall, & Sarge, 1996). Other possible influences on mood include weather, temperature, hormonal cycles, genetic temperament, sleep, food, medication, and lighting (Thayer, 1989).

MEASURING AFFECT

Measuring user affect can be valuable both as a component of usability testing and as an interface technique. When evaluating interfaces, affective information provides insight into what a user is feeling—the fundamental basis of liking and other sentiments. Within an interface, knowledge of a user's affect provides useful feedback regarding the degree to which a user's

goals are being met, enabling dynamic and intelligent adaptation. In particular, social interfaces (including character-based interfaces) must have the ability to recognize and respond to emotion in users to execute effectively real-world interpersonal interaction strategies (Picard, 1997a).

Neurological Responses

The brain is the most fundamental source of emotion. The most common way to measure neurological changes is the electroencephalogram (EEG). In a relaxed state, the human brain exhibits an alpha rhythm, which can be detected by EEG recordings taken through sensors attached to the scalp. Disruption of this signal (alpha blocking) occurs in response to novelty, complexity, and unexpectedness, as well as during emotional excitement and anxiety (Frijda, 1986). EEG studies have further shown that positive/approach-related emotions lead to greater activation of the left anterior region of the brain, while negative/avoidance-related emotions lead to greater activation of the right anterior region (Davidson, 1992; see also Heller, 1990). Indeed, when one flashes a picture to either the left or the right of where a person is looking, the viewer can identify a smiling face more quickly when it is flashed to the left hemisphere and a frowning face more quickly when it is flashed to the right hemisphere (Reuter-Lorenz & Davidson, 1981). Current EEG devices, however, are fairly clumsy and obstructive, rendering them impractical for most HCI applications. Recent advances in magneto resonance imaging (MRI) offer great promise for emotion monitoring, but are currently unrealistic for HCI because of their expense, complexity, and form factor.

Autonomic Activity

Autonomic activity has received considerable attention in studies of emotion, in part due to the relative ease in measuring certain components of the autonomic nervous system (ANS), including heart rate, blood pressure, blood-pulse volume, respiration, temperature, pupil dilation, skin conductivity, and more recently, muscle tension (as measured by electromyography (EMG)). However, the extent to which emotions can be distinguished on the basis of autonomic activity alone remains a hotly debated issue (see Ekman & Davidson, 1994, ch. 6; Levenson, 1988). On the one end are those, following in the Jamesian tradition (James, 1884), who believe that each emotion has a unique autonomic signature—technology is simply not advanced enough yet to fully detect these differentiators. On the other extreme, there are those, following Cannon (1927), who contended that all emotions are accompanied by the same state of nonspecific autonomic (sympathetic) arousal, which varies only in magnitude—most commonly measured by galvanic skin response (GSR), a measure of skin conductivity (Schachter & Singer, 1962). This controversy has clear connections to the nature-nurture debate in emotion, described earlier, because autonomic specificity seems more probable if each emotion has a distinct biological basis, while nonspecific autonomic (sympathetic) arousal seems more likely if differentiation among emotions is based mostly on cognition and social learning.

Though the debate is far from resolved, certain measures have proven reliable at distinguishing among "basic emotions." Heart rate, for example, increases most during fear, followed by anger, sadness, happiness, surprise, and finally disgust, which shows almost no change in heart rate (Cacioppo, Bernston, Klein, & Poehlmann, 1997; Ekman, Levenson, & Friesen, 1983; Levenson, Ekman, & Friesen, 1990). Heart rate also generally increases during excitement, mental concentration, and "upon the presentation of intense sensory stimuli" (Frijda, 1986). Decreases in heart rate typically accompany relaxation, attentive visual and audio observation, and the processing of pleasant stimuli (Frijda, 1986). As is now common knowledge, blood pressure increases during stress and decreases during relaxation. Cacioppo et al. (2000) further observed that anger increases diastolic blood pressure to the greatest degree, followed by fear, sadness, and happiness. Anger is further distinguished from fear by larger increases in blood pulse volume, more nonspecific skin conductance responses, smaller increases in cardiac output, and other measures indicating that "anger appears to act more on the vasculature and less on the heart than does fear" (Cacioppo et al., 1997). Results using other autonomic measures are less reliable.

Combined measures of multiple autonomic signals show promise as components of an emotion recognition system. Picard, Vyzas, and Healey (in press), for example, achieved 81% percent recognition accuracy on eight emotions through combined measures of respiration, blood pressure volume, and skin conductance, as well as facial muscle tension (to be discussed in the next subsection). Many autonomic signals can also be measured in reasonably nonobstructive ways (i.e., through user contact with mice and keyboards; (Picard, 1997a).

However, even assuming that we could distinguish among all emotions through autonomic measures, it is not clear that we should. In real-world social interactions, humans have at least partial control over what others can observe of their emotions. If another person, or a computer, is given direct access to users' internal states, users may feel overly vulnerable, leading to stress and distraction. Such personal access could also be seen as invasive, compromising trust. It may, therefore, be more appropriate to rely on measurement of the external signals of emotion (discussed next).

Facial Expression

Facial expression provides a fundamental means by which humans detect emotion. Table 4.1 describes characteristic facial features of six basic emotions (Ekman & Friesen, 1975; Rosenfeld, 1997). Endowing computers with the ability to recognize facial expressions, through pattern recognition of captured images, have proven to be a fertile area of research (Essa & Pentland, 1997; Lyons, Akamatsu, Kamachi, & Gyoba, 1998; Martinez, 2000; Yacoob & Davis, 1996); for recent reviews, see Cowie et al., 2001; Lisetti & Schiano, 2000; Tian, Kanade, & Cohn, 2001). Ekman and Friesen's (1977) Facial Action Coding System (FACS), which identifies a highly specific set of muscular movements for each emotion, is one of the most widely accepted foundations for facial-recognition systems (Tian et al., 2001). In many systems, recognition accuracy can reach as high as 90%–98% on a small set of basic emotions. However, current

TABLE 4.1. Facial Cues and Emotion

Emotion	Observed Facial Cues
Surprise	Brows raised (curved and high)
	Skin below brow stretched
	Horizontal wrinkles across forehead
	Eyelids opened and more of the white of the eye is visible
	Jaw drops open without tension or stretching of the mouth
Fear	Brows raised and drawn together
	Forehead wrinkles drawn to the center
	Upper eyelid is raised and lower eyelid is drawn up
	Mouth is open
	Lips are slightly tense or stretched and drawn back
Disgust	Upper lip is raised
	Lower lip is raised and pushed up to upper lip or it is lowered
	Nose is wrinkled
	Cheeks are raised
	Lines below the lower lid, lid is pushed up but not tense
	Brows are lowered
Anger	Brows lowered and drawn together
	Vertical lines appear between brows
	Lower lid is tensed and may or may not be raised
	Upper lid is tense and may or may not be lowered due to brows' action
	Eyes have a hard stare and may have a bulging appearance
	Lips are either pressed firmly together with corners straight or down or open, tensed in a squarish shape
	Nostrils may be dilated (could occur in sadness too) unambiguous only if registered in all three facial areas
Happiness	Corners of lips are drawn back and up
	Mouth may or may not be parted with teeth exposed or not
	A wrinkle runs down from the nose to the outer edge beyond lip corners
	Cheeks are raised
	Lower eyelid shows wrinkles below it, and may be raised but not tense
	Crow's-feet wrinkles go outward from the outer corners of the eyes
Sadness	Inner corners of eyebrows are drawn up
	Skin below the eyebrow is triangulated, with inner corner up
	Upper lid inner corner is raised
	Corners of the lips are drawn or lip is trembling

recognition systems are tested almost exclusively on "produced" expressions (i.e., subjects are asked to make specific facial movements or emotional expressions), rather than natural expressions resulting from actual emotions. The degree of accuracy that can be achieved on more natural expressions of emotion remains unclear. Further, "not all . . . emotions are accompanied by visually perceptible facial action" (Cacioppo et al., 1997).

An alternate method for facial expression recognition, capable of picking up both visible and extremely subtle movements of facial muscles, is facial electromyography (EMG). EMG signals, recorded through small electrodes attached to the skin, have proven most successful at detecting positive versus negative emotions and show promise in distinguishing among basic emotions (Cacioppo et al., 2000). Though the universality (and biological basis) of facial expression is also debated, common experience tells us that, at least within a culture, facial expressions are reasonably consistent. Nonetheless, individual differences may also be important, requiring recognition systems to adapt to a specific user for greatest accuracy. Gestures can also be recognized with technologies similar to those for facial-expression recognition, but the connection between gesture and emotional state is less distinct, in part due to the greater influence of personality (Cassell & Thorisson, in press; Collier, 1985).

Voice

Voice presents yet another opportunity for emotion recognition (see Cowie et al., 2001 for an extensive review). Emotional arousal is the most readily discernible aspect of vocal communication, but voice can also provide indications of valence and specific emotions through acoustic properties such as pitch range, rhythm, and amplitude or duration changes (Ball & Breese, 2000; Scherer, 1989). A bored or sad user, for example, will typically exhibit slower, lower-pitched speech, with little high frequency energy, while a user experiencing fear, anger, or joy will speak faster and louder, with strong high-frequency energy and more explicit enunciation (Picard, 1997a). Murray and Arnott (1993) provided a detailed account of the vocal effects associated with several basic emotions (see Table 4.2). Though few systems have been built for automatic emotion recognition through speech, Banse and Scherer (1996) have demonstrated the feasibility of such systems. Cowie and Douglas-Cowie's ACCESS system (Cowie & Douglas-Cowie, 1996) also presents promise (Cowie et al., 2001).

Self-Report Measures

A final method for measuring a user's affective state is to ask questions. Post-interaction questionnaires, in fact, currently

TABLE 4.2. Voice and Emotion

	Fear	Anger	Sadness	Happiness	Disgust
Speech rate	Much faster	Slightly faster	Slightly slower	Faster or slower	Very much slower
Pitch average	Very much higher	Very much higher	Slightly lower	Much higher	Very much lower
Pitch range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider
Intensity	Normal	Higher	Lower	Higher	Lower
Voice quality	Irregular voicing	Breathy chest tone	Resonant	Breathy blaring	Grumbled chest tone
Pitch changes	Normal	Abrupt on stressed syllables	Downward inflections	Smooth upward inflections	Wide downward terminal inflections
Articulation	Precise	Tense	Slurring	Normal	Normal

serve as the primary method for ascertaining emotion, mood, and sentiment during an interaction. However, in addition to the standard complexities associated with self-report measures (such as the range of social desirability effects), measuring affect in this way presents added challenges. To begin with, questionnaires are capable of measuring only the *conscious experience* of emotion and mood. Much of affective processing, however, resides in the limbic system and in nonconscious processes. Although it is debatable whether an emotion can exist without any conscious component at all, a mood surely can. Further, questions about emotion, and often those about mood, refer to past affective states and thus rely on imperfect and potentially biased memory. Alternatively, asking a user to report on an emotion as it occurs requires interruption of the experience. In addition, emotions and moods are often difficult to describe in words. Finally, questions about sentiment, although the most straightforward given their predictive nature, are potentially affected by when they are asked (both because of current mood and memory degradation). Nevertheless, self-report measures are the most direct way to measure sentiment and a reasonable alternative to direct measures of emotion and mood (which currently remain in the early stages of development).

Several standard questionnaires exist for measuring affect (Plutchik & Kellerman, 1989, Chapter 1–3). The most common approach presents participants with a list of emotional adjectives and asks how well each describes their affective state. Izard's (1972) Differential Emotion Scale (DES), for example, includes 24 emotional terms (such as delighted, scared, happy, and astonished) that participants rate on seven-point scales indicating the degree to which they are feeling that emotion (from "not at all" to "extremely"). McNair, Lorr, and Droppleman's (1981) Profile of Mood States (POMS) is a popular adjective-based measure of mood. Researchers have created numerous modifications of these standard scales (Desmet, Hekkert, & Jacobs, 2000, presented a unique nonverbal adaptation), and many current usability questionnaires include at least some adjective-based affect assessment items (i.e., the Questionnaire for User Interface Satisfaction (QUIS) (Chin, Diehl, & Norman, 1988)).

A second approach to questionnaire measurement of affect derives from dimensional theories of emotion and mood. Many researchers argue that two dimensions—arousal (activation) and valence (pleasant/unpleasant)—are nearly sufficient to describe the entire space of conscious emotional experience (Feldman, Barrett, & Russell, 1999). Lang (1995), for example, presented an interesting measurement scheme where subjects rate the arousal and valence of their current affective state by selecting among pictorial representations (rather than the standard number/word representation of degree). Watson, Clark, and Tellegen's (1988) Positive and Negative Affect Schedule (PANAS) is a popular dimensional measure of mood. Finally, to measure emotion as it occurs, with minimum interruption, some researchers have asked subjects to push one of a small number of buttons indicating their current emotional reaction during presentation of a stimulus (i.e., one button each for positive, negative, and neutral response (Breckler & Berman, 1991)).

Affect Recognition by Users

Computers are not the only (potential) affect recognizers in human-computer interactions. When confronted with an interface—particularly a social or character-based interface—users constantly monitor cues to the affective state of their interaction partner, the computer (though often nonconsciously; see Reeves & Nass, 1996). Creating natural and efficient interfaces requires not only recognizing emotion in users, but also expressing emotion. Traditional media creators have known for a long time that portrayal of emotion is a fundamental key to creating the "illusion of life" (Jones, 1990; Thomas & Johnson, 1981; for discussions of believable agents and emotion, see, i.e., Bates, 1994; Maldonado, Picard, & Hayes-Roth, 1998).

Facial expression and gesture are the two most common ways to manifest emotion in screen-based characters (Cassell et al., 2000; Kurlander, Skelly, & Salesin, 1996). Though animated expressions lack much of the intricacy found in human expressions, users are nonetheless capable of distinguishing emotions in animated characters (Cassell et al., 2000; Schiano, Ehrlich, Rahardja, & Sheridan, 2000). As with emotion recognition, Ekman and Friesen's (1977) Facial Action Coding System (FACS) is a commonly used and well-developed method for constructing affective expressions. One common strategy for improving accurate communication with animated characters is to exaggerate expressions, but whether this leads to corresponding exaggerated assumptions about the underlying emotion has not been studied.

Characters that talk can also use voice to communicate emotion (Nass & Gong, 2000). Prerecorded utterances are easily infused with affective tone, but are fixed and inflexible. Cahn (1990) have successfully synthesized affect-laden speech using a text-to-speech (TTS) system coupled with content-sensitive rules regarding appropriate acoustic qualities (including pitch, timing, and voice quality; see also Nass, Foehr, & Somoza, 2000). Users were able to distinguish among six different emotions with about 50% accuracy, which is impressive considering that people are generally only 60% accurate in recognizing affect in *human* speech (Scherer, 1981).

Finally, characters can indicate affective state verbally through word and topic choice, as well as explicit statements of affect (i.e., "I'm happy"). Characters, whose nonverbal and verbal expressions are distinctly mismatched, however, may be seen as awkward or even untrustworthy. In less extreme mismatched cases, recent evidence suggests that users will give precedence to nonverbal cues in judgments about affect (Nass et al., 2000). This finding is critical for applications in which characters/agents mediate interpersonal communication (i.e., in virtual worlds or when characters read email to a user), because the affective tone of a message may be inappropriately masked by the character's affective state. Ideally, in such computer-mediated communication contexts, emotion would be encoded into the message itself, either through explicit tagging of the message with affect, through natural language processing of the message, or through direct recognition of the sender's affective state during message composition (i.e., using autonomic nervous system or facial expression measures). Mediator characters could then

display the appropriate nonverbal cues to match the verbal content of the message.

OPEN QUESTIONS

Beyond the obvious need for advancements in affect recognition and manifestation technology, it is our opinion that there are five important and remarkably unexplored areas for research in emotion and HCI:

1. *With which emotion should* HCI *designers be most concerned*?

Which emotion(s) should interface designers address first? The basic emotions, to the extent that they exist and can be identified, have the advantage of similarity across cultures and easy discriminability. Thus, designs that attempt to act upon or manipulate these dimensions may be the simplest to implement. However, within these basic emotions, little is known about their relative manipulability or manifestability—particularly within the HCI context—or their relative impact on individuals' attitudes and behaviors. Once one moves beyond the basic emotions, cultural and individual differences introduce further problems and opportunities.

2. *When and how should interfaces attempt to directly address users' emotions and basic needs* (*vs. application-specific goals*)?

If one views a computer or an interface merely as a tool, then interface design should solely focus on application-specific goals, assessed by such metrics as efficiency, learnability, and accuracy. However, if computers and interfaces are understood as a *medium*, then it becomes important to think about both uses and *gratifications* (Katz, Blumler, & Gurevitch, 1974; Rosengren, 1974; Rubin, 1986); that is, the more general emotional and basic needs that users bring to any interaction. Notions of "infotainment" or "edutainment" indicate one category of attempts to balance task and affect. However, there is little understanding of how aspects of interfaces that directly manipulate users' emotions compliment, undermine, or are orthogonal to aspects of interfaces that specifically address users' task needs.

3. *How accurate must emotion recognition be to be useful as an interface technique*?

Although humans are not highly accurate emotion detectors—the problem of "receiving accuracy" (Picard, 1997a, p. 120)—they nonetheless benefit from deducing other's emotions and acting on those deductions (Goleman, 1995). Clearly, however, a minimum threshold of accuracy is required before behavior based on emotion induction is appropriate. Very little is known about the level of confidence necessary before an interface can effectively act on a user's emotional state.

4. *When and how should users be informed that their affective states are being monitored and adapted to*?

When two people interact, there is an *implicit* assumption that each person is monitoring the other's emotional state and responding based on that emotional state. However, an explicit statement of this fact would be highly disturbing: "To facilitate our interaction, I will carefully and constantly monitor everything you say and do to discern your emotional state and respond based on that emotional state" or "I have determined that you are sad; I will now perform actions that will make you happier." However, when machines acquire and act upon information about users without making that acquisition and adaptation explicit, there is often a feeling of "surreptitiousness" or "manipulation." Furthermore, if emotion monitoring and adapting software are desired by consumers, there are clearly incentives for announcing and marketing these abilities. Because normal humans only exhibit implicit monitoring, the psychological literature is silent on the psychological and performance implications for awareness of emotional monitoring and adaptation.

5. *How does emotion play out in computer-mediated communication* (CMC)?

This chapter has focused on the direct relationship between the user and the interface. However, computers also are used to *mediate* interactions between people. In face-to-face encounters, affect not only creates richer interaction, but also helps to disambiguate meaning, allowing for more effective communication. Little is known, however, about the psychological effects of mediated affect, or the optimal strategies for encoding and displaying affective messages (see Maldonado & Picard, 1999; Rivera, Cooke, & Bauhs, 1996).

CONCLUSION

Though much progress has been made in the domain of "affective computing" (Picard, 1997a), more work is clearly necessary before interfaces that incorporate emotion recognition and manifestation can reach their full potential. Nevertheless, a careful consideration of affect in interaction design and testing can be instrumental in creating interfaces that are both efficient and effective, as well as enjoyable and satisfying. Designers and theorists, for even the simplest interfaces, are well advised to thoughtfully address the intimate and far-reaching linkages between emotion and HCI.

ACKNOWLEDGMENTS

James Gross, Heidi Maldonado, Roz Picard, and Don Roberts provided extremely detailed and valuable insights. Sanjoy Banerjee, Daniel Bochner, Dolores Canamero, Pieter Desmet, Ken Fabian, Michael J. Lyons, George Marcus, Laurel Margulis, Byron Reeves, and Aaron Sloman provided useful suggestions.

References

Averill, J. R. (1980). A contructionist view of emotion. In R. Plutchik & H. Kellerman (Eds.), *Emotion: Theory, research, and experience* (Vol. 1, pp. 305–339). New York: Academic Press.

Ball, G., & Breese, J. (2000). Emotion and personality in conversational agents. In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 189–219). Cambridge, MA: The MIT Press.

Banse, R., & Scherer, K. (1996). Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psychology, 70*(3), 614–636.

Barrett, L. F., & Russell, J. A. (1999). The structure of current affect: Controversies and emerging consensus. *Current Directions in Psychological Science, 8*(1), 10–14.

Bates, J. (1994). The role of emotions in believable agents. *Communications of the ACM, 37*(7), 122–125.

Bates, J., Loyall, A. B., & Reilly, W. S. (1994). An architecture for action, emotion, and social behavior, *Artificial social systems: Fourth European workshop on modeling autonomous agents in a multi-agent world*. Berlin, Germany: Springer-Verlag.

Bishop, D., Jacks, H., & Tandy, S. B. (1993). The Structure of Temperament Questionnaire (STQ): Results from a U.S. sample. *Personality & Individual Differences, 14*(3), 485–487.

Bower, G. H. (1992). How might emotions affect learning? In C. Svenåke (Ed.), *The handbook of emotion and memory: Research and theory* (pp. 3–31). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bower, G. H., & Forgas, J. P. (2000). Affect, memory, and social cognition. In E. Eich, J. F. Kihlstrom, G. H. Bower, J. P. Forgas, & P. M. Niedenthal (Eds.), *Cognition and emotion* (pp. 87–168). Oxford, U.K.: Oxford University Press.

Bower, G. H., Gilligan, S. G., & Monteiro, K. P. (1981). Selectivity of learning caused by affective states. *Journal of Experimental Psychology: General, 110*, 451–473.

Bozinovski, S., & Bozinovska, L. (2001). Self-learning agents: A connectionist theory of emotion based on crossbar value judgment. *Cybernetics and Systems, 32*, 5–6.

Breckler, S. T., & Berman, J. S. (1991). Affective responses to attitude objects: Measurement and validation. *Journal of Social Behavior and Personality, 6*(3), 529–544.

Cacioppo, J. T., Bernston, G. G., Klein, D. J., & Poehlmann, K. M. (1997). Psychophysiology of emotion across the life span. *Annual Review of Gerontology and Geriatrics, 17*, 27–74.

Cacioppo, J. T., Bernston, G. G., Larsen, J. T., Poehlmann, K. M., & Ito, T. A. (2000). The psychophysiology of emotion. In M. Lewis & J. M. Haviland-Jones (Eds.), *Handbook of emotions* (2nd ed., pp. 173–191). New York: The Guilford Press.

Cahn, J. E. (1990). The generation of affect in sythesized speech. *Journal of the American Voice I/O Society, 8*, 1–19.

Cannon, W. B. (1927). The James-Lange theory of emotions: A critical examination and an alternate theory. *American Journal of Psychology, 39*, 106–124.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. (Eds.). (2000). *Embodied conversational agents*. Cambridge, MA: MIT Press.

Cassell, J., & Thorisson, K. (in press). The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Journal of Applied Artificial Intelligence*.

Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface, *Proceedings of CHI '88 human factors in computing systems* (pp. 213–218). New York: ACM Press.

Clore, G. C. (1994). Why emotions are felt. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion: Fundamental questions* (pp. 103–111). New York: Oxford University Press.

Clore, G. C., & Gasper, K. (2000). Feeling is believing: Some affective influences on belief. In N. H. Frijda, A. S. R. Manstead, & S. Bem (Eds.), *Emotions and beliefs: How feelings influence thoughts* (pp. 10–44). Cambridge, UK: Cambridge University Press.

Clore, G. C., Wyer, R. S., Jr., Diened, B., Gasper, K., Gohm, C., & Isbell, L. (2001). Affective feelings as feedback: Some cognitive consequences. In L. L. Martin & G. C. Clore (Eds.), *Theories of mood and cognition: A user's handbook* (pp. 63–84). Mahwah, NJ: Lawrence Erlbaum Associates.

Collier, G. (1985). *Emotional expression*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cowie, R., & Douglas-Cowie, E. (1996). *Automatic statistical analysis of the signal and posidic signs of emotion in speech.* Paper presented at the Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP-96), Oct 3–6, New Castle, DE.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., & Taylor, J. G. (2001). Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine, 18*(1), 32–80.

Damasio, A. R. (1994). *Descartes' error: Emotion, reason, and the human brain.* New York: Putnam Publishing Group.

Darwin, C. (1872/1998). *The expression of the emotions in man and animals*. London: HarperCollins.

Davidson, R. J. (1992). Anterior cerebral asymmetry and the nature of emotion. *Brain and Cognition, 20*, 125–151.

Davidson, R. J. (1994). On emotion, mood, and related affective constructs. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion* (pp. 51–55). New York: Oxford University Press.

Detenber, B. H., & Reeves, B. (1996). A bio-informational theory of emotion: Motion and image size effects on viewers. *Journal of Communication, 46*(3), 66–84.

Duncker, K. (1945). On problem-solving. *Psychological Monographs, 58*(Whole No. 5).

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion, 6*(3/4), 169–200.

Ekman, P. (1994). All emotions are basic. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion: Fundamental questions* (pp. 7–19). New York: Oxford University Press.

Ekman, P., & Davidson, R. J. (Eds.). (1994). *The nature of emotion*. New York: Oxford University Press.

Ekman, P., & Friesen, W. V. (1975). *Unmasking the face*. Englewood Cliffs, NJ: Prentice-Hall.

Ekman, P., & Friesen, W. V. (1977). *Manual for the Facial Action Coding System.* Palo Alto: Consulting Psychologist Press.

Ekman, P., Levenson, R. W., & Friesen, W. V. (1983). Autonomic nervous system activity distinguishes among emotions. *Science, 221*, 1208–1210.

Elliott, C., & Brzezinski, J. (1998, Summer). Autonomous agents and synthetic characters. *AI magazine*, 13–30.

Ellis, H. C., & Moore, B. A. (1999). Mood and memory. In T. Dalgleish & M. J. Power (Eds.), *Handbook of cognition and emotion* (pp. 193–210). New York: John Wiley & Sons.

Ellsworth, P. C. (1994). Some reasons to expect universal antecedents of emotion. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion: Fundamental questions* (pp. 150–154). New York: Oxford University Press.

Erber, R., & Erber, M. W. (2001). Mood and processing: A view from a self-regulation perspective. In L. L. Martin & G. C. Clore (Eds.), *Theories of mood and cognition: A user's handbook* (pp. 63–84). Mahwah, NJ: Lawrence Erlbaum Associates.

Essa, I. A., & Pentland, A. P. (1997). Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 757–763.

Feldman Barrett, L., & Russell, J. A. (1999). The structure of current affect: Controversies and emerging consensus. *Current Directions in Psychological Science, 8*(1), 10–14.

Fogg, B. J. (1998). Charismatic computers: Creating more likable and persuasive interactive technologies by leveraging principles from social psychology. *Dissertation Abstracts International Section A: Humanities & Social Sciences, 58*(7-A), 2436.

Forgas, J. P. (1995). Mood and judgment: The Affect Infusion Model (AIM). *Psychological Bulletin, 117*, 39–66.

Friedman, B., & Kahn, P. H., Jr. (1997). Human agency and responsible computing: Implications for computer system design. In B. Friedman (Ed.), *Human values and the design of computer technology* (pp. 221–235). Stanford, CA: CSLI Publications.

Frijda, N. H. (1986). *The emotions*. Cambridge, UK: Cambridge University Press.

Frijda, N. H. (1988). The laws of emotion. *American Psychologist, 43*(5), 349–358.

Frijda, N. H. (1994). Varieties of affect: Emotions and episodes, moods, and sentiments. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion* (pp. 59–67). New York: Oxford University Press.

Gardano, A. C. (1986). Cultural influence on emotional response to color: A research study comparing Hispanics and non-Hispanics. *American Journal of Art Therapy, 23*, 119–124.

Gerard, R. (1957). *Differential effects of colored lights on psychophysiological functions.* Unpublished doctoral dissertation, University of California, Los Angeles, CA.

Gerard, R. (1958, July 13). Color and emotional arousal [Abstract]. *American Psychologist, 13*, 340.

Goleman, D. (1995). *Emotional intelligence*. New York: Bantam Books.

Gregory, A. H., Worrall, L., & Sarge, A. (1996). The development of emotional responses to music in young children. *Motivation and Emotion, 20*(4), 341–348.

Gross, J. J. (1998). Antecedent- and response-focused emotion regulation: Divergent consequences for experience, expression, and physiology. *Journal of Personality and Social Psychology, 74*, 224–237.

Gross, J. J. (1999). Emotion and emotion regulation. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research* (2nd ed., pp. 525–552). New York: Guildford.

Hatfield, E., Cacioppo, J. T., & Rapson, R. L. (1994). *Emotional contagion*. Paris/New York: Editions de la Maison des Sciences de l'Homme and Cambridge University Press (jointly published).

Heller, W. (1990). The neuropsychology of emotion: Developmental patterns and implications for psychpathology. In N. L. Stein, B. Leventhal, & T. Trabasso (Eds.), *Psychological and biological approaches to emotion* (pp. 167–211). Hillsdale, NJ: Lawrence Erlbaum Associates.

Henry, D. L., & Jacobs, K. W. (1978). Color eroticism and color preference. *Perceptual and Motor Skills, 47*, 106.

Heuer, F., & Reisberg, D. (1992). Emotion, arousal, and memory for detail. In C. Sven-Åke (Ed.), *The handbook of emotion and memory: Research and theory* (pp. 3–31). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hirt, E. R., Melton, R. J., McDonald, H. E., & Harackiewicz, J. M. (1996). Processing goals, task interest, and the mood-performance relationship: A mediational analysis. *Journal of Personality and Social Psychology, 71*, 245–261.

Isen, A. M. (2000). Positive affect and decision making. In M. Lewis & J. M. Haviland-Jones (Eds.), *Handbook of emotions* (2nd ed., pp. 417–435). New York: The Guilford Press.

Isen, A. M., Daubman, K. A., & Nowicki, G. P. (1987). Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology, 52*(6), 1122–1131.

Isen, A. M., Rosenzweig, A. S., & Young, M. J. (1991). The influence of positive affect on clinical problem solving. *Medical Decision Making, 11*(3), 221–227.

Izard, C. E. (1972). *Patterns of emotions*. New York: Academic Press.

Izard, C. E. (1992). Basic emotions, relations among emotions, and emotion-cognition relations. *Psychological Review, 99*(3), 561–565.

James, W. (1884). What is an emotion? *Mind, 9*, 188–205.

Jones, C. (1990). *Chuck amuck: The life and times of an animated cartoonist*. New York: Avon Books.

Katz, E., Blumler, J. G., & Gurevitch, M. (1974). Utilization of mass communication by the individual. In J. G. Blumler & E. Katz (Eds.), *The uses of mass communcations: Current perspectives on gratifications research* (pp. 19–32). Beverly Hills, CA: Sage.

Klein, J., Moon, Y., & Picard, R. W. (1999). *This computer responds to user frustration* (pp. 242–243). Paper presented at the Human factors in computing systems: CHI'99 extended abstracts, New York.

Kleinginna, P. R., Jr., & Kleinginna, A. M. (1981). A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion, 5*(4), 345–379.

Kurlander, D., Skelly, T., & Salesin, D. (1996). *Comic chat.* Paper presented at the Proceedings of SIGGRAPH'96: International conference on computer graphics and interactive techniques, Aug 4–9, New York.

Lang, P. J. (1995). The emotion probe. *American Psychologist, 50*(5), 372–385.

Lazarus, R. S. (1991). *Emotion and adaptation*. New York: Oxford University Press.

LeDoux, J. E. (1995). Emotion: Clues from the brain. *Annual Review of Psychology, 46*, 209–235.

LeDoux, J. E. (1996). *The emotional brain*. New York: Simon & Schuster.

LeDoux, J. E., & Phelps, E. A. (2000). Emotional networks in the brain. In M. Lewis & J. M. Haviland-Jones (Eds.), *Handbook of emotions* (pp. 157–172). New York: The Guilford Press.

Levenson, R. W. (1988). Emotion and the autonomic nervous system: A prospectus for research on autonomic specificity. In H. Wagner (Ed.), *Social psychophysiology: Perspectives on theory and clinical applications* (pp. 17–42). London: Wiley.

Levenson, R. W., Ekman, P., & Friesen, W. V. (1990). Voluntary facial action generates emotion-specific autonomic nervous system activity. *Psychophysiology, 27*, 363–384.

Levy, B. I. (1984). Research Into the psychological meaning of color. *American Journal of Art Therapy, 23*, 58–62.

Lisetti, C. L., & Schiano, D. J. (2000). Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect [Special issue on Facial Information Processing: A Multidisciplinary Perspective]. *Pragmatics and Cognition, 8*(1), 185–235.

Lyons, M., Akamatsu, S., Kamachi, M., & Gyoba, J. (1998). Coding facial expressions with gabor wavelets, *Proceedings of the Third IEEE international conference on automatic face and gesture recognition* (pp. 200–205). New York: Avon Books.

Malamuth, N. (1996). Sexually explicit media, gender differences, and evolutionary theory. *Journal of Communication, 46*, 8–31.

Maldonado, H., & Picard, A. (1999). The Funki Buniz playground: Facilitating multi-cultural affective collaborative play [Abstract]. *CHI99 extended abstracts* (pp. 328–329).

Maldonado, H., Picard, A., & Hayes-Roth, B. (1998). Tigrito: A high affect virtual toy. *CHI98 Summary* (pp. 367–368).

Martinez, A. M. (2000). *Recognition of partially occluded and/or imprecisely localized faces using a probabilistic approach.* Paper presented at the Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR '2000). June 13–15, Hilton Head, SC, USA.

Maslow, A. H. (1968). *Toward a psychology of being*. New York: D. Van Nostrand Company.

McNair, D. M., Lorr, M., & Droppleman, L. F. (1981). *Manual of the Profile of Mood States*. San Diego, CA: Educational and Industrial Testing Services.

Morkes, J., Kernal, H. K., & Nass, C. (2000). Effects of humor in task-oriented human-computer interaction and computer-mediated communication: A direct test of SRCT theory. *Human-Computer Interaction, 14*(4), 395–435.

Murphy, S. T., & Zajonc, R. B. (1993). Affect, cognition, and awareness: Affective priming with suboptimal and optimal stimulus. *Journal of Personality and Social Psychology, 64*, 723–739.

Murray, I. R., & Arnott, J. L. (1993). Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *Journal Acoustical Society of America, 93*(2), 1097–1108.

Murray, N., Sujan, H., Hirt, E. R., & Sujan, M. (1990). The influence of mood on categorization: A cognitive flexibility interpretation. *Journal of Personality and Social Psychology, 59*, 411–425.

Nass, C., Foehr, U., & Somoza, M. (2000). *The effects of emotion of voice in synthesized and recorded speech*. Unpublished manuscript.

Nass, C., & Gong, L. (2000). Speech interfaces from an evolutionary perspective. *Communications of the ACM*, *43*(9), 36–43

Neese, R. M. (1990). Evolutionary explanations of emotions. *Human Nature, 1*(3), 261–289.

Neumann, R., & Strack, F. (2000). "Mood contagion": The automatic transfer of mood between persons. *Journal of Personality and Social Psychology, 79*(2), 211–223.

Newhagen, J., & Reeves, B. (1991). Emotion and memory responses to negative political advertising. In F. Biocca (Ed.), *Televions and political advertising: Psychological processes* (pp. 197–220). Hillsdale, NJ: Lawrence Erlbaum.

Newhagen, J., & Reeves, B. (1992). This evening's bad news: Effects of compelling negative television news images on memory. *Journal of Communication, 42*, 25–41.

Niedenthal, P. M., Setterlund, M. B., & Jones, D. E. (1994). Emotional organization of perceptual memory. In P. M. Niedenthal & S. Kitayama (Eds.), *The heart's eye* (pp. 87–113). San Diego, CA: Academic Press, Inc.

Norman, D. (1990). *The design of everyday things*. Garden City, NJ: Doubleday.

Oatley, K., & Johnson-Laird, P. N. (1987). Towards a cognitive theory of emotions. *Cognition and Emotion, 1*(1), 29–50.

Ortony, A., Clore, G. C., & Collins, A. (1988). *The cognitive structure of emotions*. Cambridge, MA: Cambridge University Press.

Ortony, A., & Turner, T. J. (1990). What's basic about emotions. *Psychological Review, 97*(3), 315–331.

Panksepp, J. (1992). A critical role for "affective neuroscience" in resolving what is basic about basic emotions. *Psychological Review, 99*(3), 554–560.

Parrott, G. W., & Spackman, M. P. (2000). Emotion and memory. In M. Lewis & J. M. Haviland-Jones (Eds.), *Handbook of emotions* (2nd ed., pp. 476–490). New York: The Guilford Press.

Pecjak, V. (1970). Verbal synthesis of colors, emotions and days of the week. *Journal of Verbal Learning and Verbal Behavior, 9*, 623–626.

Peretti, P. O. (1974). Color model associations in young adults. *Perceptual and Motor Skills, 39*, 715–718.

Picard, R. W. (1997a). *Affective computing*. Cambridge, MA: The MIT Press.

Picard, R. W. (1997b). Does HAL cry digital tears? Emotions and computers. In D. G. Stork (Ed.), *Hal's Legacy: 2001's Computer as Dream and Reality* (pp. 279–303). Cambridge, MA: The MIT Press.

Picard, R. W., Vyzas, E., & Healey, J. (in press). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Plutchik, R., & Kellerman, H. (Eds.). (1989). *Emotion: Theory, research, and experience* (Vol. 4: The Measurement of Emotions). San Diego, CA: Academic Press, Inc.

Reeves, B., Biocca, F., Pan, Z., Oshagan, H., & Richards, J. (1989). *Unconscious processing and priming with pictures: Effects on emotional attributions about people on television*. Unpublished manuscript.

Reeves, B., & Nass, C. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. New York: Cambridge University Press.

Reeves, B., Newhagen, J., Maibach, E., Basil, M. D., & Kurz, K. (1991). Negative and positive television messages: Effects of message type and message content on attention and memory. *American Behavioral Scientist, 34*, 679–694.

Reuter-Lorenz, P., & Davidson, R. J. (1981). Differential contributions of the two cerebral hemispheres to the perception of happy and sad faces. *Neuropsychologia, 19*(4), 609–613.

Rivera, K., Cooke, N. J., & Bauhs, J. A. (1996). The effects of emotional icons on remote communication. *CHI '96 interactive posters*, 99–100.

Roseman, I. J., Antoniou, A. A., & Jose, P. E. (1996). Appraisal determinants of emotions: Constructing a more accurate and comprehensive theory. *Cognition and Emotion, 10*(3), 241–277.

Rosenfeld, A. (1997). Eyes for computers: How HAL could "see". In D. G. Stork (Ed.), *Hal's legacy: 2001's Computer as dream and reality* (pp. 211–235). Cambridge, MA: The MIT Press.

Rosengren, K. E. (1974). Uses and gratifications: A paradigm outlined. In J. G. Blumler & E. Katz (Eds.), *The uses of mass communications: Current perspectives on gratifications research* (pp. 269–286). Beverly Hills, CA: Sage.

Rubin, A. M. (1986). Uses, gratifications, and media effects research. In J. Bryant & D. Zillman (Eds.), *Perspectives on media effects* (pp. 281–301). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schachtel, E. J. (1943). On color and affect. *Psychiatry, 6*, 393–409.

Schachter, S., & Singer, J. E. (1962). Cognitive, social, and physiological determinants of emotional state. *Psychological Review, 69*(5), 379–399.

Scherer, K. (1981). Speech and emotional states. In J. K. Darby (Ed.), *Speech evaluation in psychiatry* (pp. 189–220). New York: Grune & Stratton.

Scherer, K. R. (1988). Criteria for emotion-antecedent appraisal: A review. In V. Hamilton, G. H. Bower, & N. H. Frijda (Eds.), *Cognitive perspectives on emotion and motivation* (pp. 89–126). Dordrecht, the Netherlands: Kluwer Academic.

Scherer, K. R. (1989). Vocal measurement of emotion. In R. Plutchik & H. Kellerman (Eds.), *Emotion: Theory, research, and experience* (Vol. 4, pp. 233–259). San Diego, CA: Academic Press, Inc.

Scheutz, M., Sloman, A., & Logan, B. (2000, November 11–12). *Emotional states and realistic agent behavior.* Paper presented at the GAME-ON 2000, Imperial College, London.

Schiano, D. J., Ehrlich, S. M., Rahardja, K., & Sheridan, K. (2000, April 1–6). *Face to interFace: Facial affect in (hu)man and machine.* Paper presented at the Human factors in computing systems: CHI'00 conference proceedings, New York.

Schwartz, N., & Bless, H. (1991). Happy and mindless, but sad and smart?: The impact of affective states on analytic reasoning. In J. P. Forgas (Ed.), *Emotion and social judgment* (pp. 55–71). Oxford, U.K.: Pergamon.

Shweder, R. A. (1994). "You're not sick, you're just in love": Emotions as an interpretive system. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotions* (pp. 32–44). New York: Oxford University Press.

Skinner, E. A. (1995). *Perceived control, motivation, & coping*. Newbury Park, CA: Sage Publications.

Thayer, R. E. (1989). *The biopsychology of mood and arousal*. New York: Oxford University Press.

Thomas, F., & Johnson, O. (1981). *The illusion of life*: *Disney animation.* New York: Abbeville Press.

Thorson, E., & Friestad, M. (1985). The effects on emotion on episodic memory for television commercials. In P. Cafferata & A. Tybout (Eds.), *Advances in consumer psychology* (pp. 131–136). Lexington, MA: Lexington.

Tian, Y.-L., Kanade, T., & Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 1–19.

Tooby, J., & Cosmides, L. (1990). The past explains the present: Emotional adaptations and the structure of ancestral environments. *Ethology and Sociobiology, 11*, 407–424.

Ucros, C. G. (1989). Modd state-dependent memory: A meta-analysis. *Cognition and Emotion, 3*(139–167).

Voelker, D. (1994). *The effects of image size and voice volume on he evlauaiton of represented faces.* Unpublished Dissertation, Stanford University, Stanford, CA.

Walters, J., Apter, M. J., & Svebak, S. (1982). Color preference, arousal, and theory of psychological reversals. *Motivation and Emotion, 6*(3), 193–215.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology, 54*, 128–141.

Wegner, D. M. (1994). Ironic processes of mental control. *Psychological Review, 101*, 34–52.

Wegner, D. M., Bargh, J. A., Gilbert, D. T., Fiske, S. T., et al. (1998). Control and automaticity in social life, *The handbook of social psychology* (4th ed., Vol. 1). Boston: McGraw-Hill.

Wexner, L. (1954). The degree to which colors (hues) are associated with mood-tones. *Journal of Applied Psychology, 38*, 432–435.

Wierzbicka, A. (1992). Talking about emotions: Semantics, culture, and cognition. *Cognition and Emotion, 6*(3/4), 285–319.

Yacoob, Y., & Davis, L. S. (1996). Recognizing human facial expresisons from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(6), 636–642.

Zillmann, D. (1991). Television viewing and physiological arousal. In J. Bryant & D. Zillman (Eds.), *Responding to the screen: Reception and reaction processes* (pp. 103–133). Hillsdale, NJ: Lawrence Erlbaum Associates.

◆5◆

COGNITIVE ARCHITECTURE

Michael D. Byrne
Rice University

What Are Cognitive Architectures? **94**
Relevance to Human-Computer Interaction **95**
Brief Look at Past Systems in HCI **96**
The Model Human Processor (MHP) and GOMS 97
Cognitive Complexity Theory (CCT) 98
CAPS 99
Soar 100
Contemporary Architectures **102**
LICAI/CoLiDeS 102
EPIC 103
ACT-R 5.0 106
Comparisons 109
The Future of Cognitive Architectures in HCI **110**
References **111**

> Designing interactive computer systems to be efficient and easy to use is important so that people in our society may realize the potential benefits of computer-based tools. . . . Although modern cognitive psychology contains a wealth of knowledge of human behavior, it is not a simple matter to bring this knowledge to bear on the practical problems of design—to build an applied psychology that includes theory, data, and knowledge. (Card, Moran, & Newell, 1983, p. vii)

Integrating theory, data, and knowledge about cognitive psychology and human performance in a way that is useful for guiding design in HCI is still not a simple matter. However, there have been significant advances since Card, Moran, and Newell (1983) wrote the previous passage. One of the key advances is the development of cognitive architectures, the subject of this chapter. The chapter will first consider what it is to be a cognitive architecture and why cognitive architectures are relevant for HCI. In order to detail the present state of cognitive architectures in HCI, it is important to consider some of the past uses of cognitive architectures in HCI research. Then, three architectures actively in use in the research community (LICAI/CoLiDeS, EPIC, and ACT-R) and their application to HCI will be examined. The chapter will conclude with a discussion of the future of cognitive architectures in HCI.

WHAT ARE COGNITIVE ARCHITECTURES?

Most dictionaries will list several different definitions for the word *architecture.* For example, dictionary.com lists among them "a style and method of design and construction," e.g., Byzantine architecture; "orderly arrangement of parts; structure," e.g., the architecture of a novel; and one from computer science: "the overall design or structure of a computer system." What, then, would something have to be to qualify as a "cognitive architecture"? It is something much more in the latter senses of the word *architecture,* an attempt to describe the overall structure and arrangement of a very particular thing, the human cognitive system. A cognitive architecture is a broad theory of human cognition based on a wide selection of human experimental data, and implemented as a running computer simulation program. Young (Gray, Young, & Kirschenbaum, 1997; Ritter & Young, 2001) defined a cognitive architecture as an embodiment of "a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task."

This idea has been a part of cognitive science since the early days of cognitive psychology and artificial intelligence, as manifested in the General Problem Solver or GPS (Newell & Simon, 1963), one of the first successful computational cognitive models. These theories have progressed a great deal since GPS and are gradually becoming broader. One of the best descriptions of the vision for this area is presented in Newell's (1990) book *Unified Theories of Cognition*. In it, Newell argued that the time has come for cognitive psychology to stop collecting disconnected empirical phenomena and begin seriously considering theoretical unification in the form of computer simulation models. Cognitive architectures are attempts to do just this.

Cognitive architectures are distinct from engineering approaches to artificial intelligence, which strive to construct intelligent computer systems by whatever technologies best serve that purpose. Cognitive architectures are designed to simulate human intelligence in a humanlike way (Newell, 1990). For example, the chess program that defeated Kasparov, Deep Blue, would not qualify as a cognitive architecture because it does not solve the problem (chess) in a humanlike way. Deep Blue uses massive search of the game space, while human experts generally look only a few moves ahead, but concentrate effectively on quality moves.

Cognitive architectures differ from traditional research in psychology in that work on cognitive architecture is integrative—that is, architectures include mechanisms to support attention, memory, problem solving, decision making, learning, and so forth. Most theorizing in psychology follows a divide-and-conquer strategy that tends to generate highly specific theories of a very limited range of phenomena; this has changed little since the 1970s (Newell, 1973). This limits the usefulness of such theories for an applied domain such as HCI, where users employ a wide range of cognitive capabilities in even simple tasks. Instead of asking, "How can we describe this isolated phenomenon?" people working with cognitive architectures can ask, "How does this phenomenon fit in with what we already know about other aspects of cognition?"

Another important feature of cognitive architectures is that they specify only the human "virtual machine," the fixed architecture. A cognitive architecture alone cannot do anything. Generally, the architecture has to be supplied with the knowledge needed to perform a particular task. The combination of an architecture and a particular set of knowledge is generally referred to as a "computational cognitive model," or just a "model." In general, it is possible to construct more than one model for any particular task. The specific knowledge incorporated into a particular model is determined by the modeler. Because the relevant knowledge must be supplied to the architecture, the knowledge engineering task facing modelers attempting to model performance on complex tasks can be formidable.

A third centrally important feature of cognitive architectures is that they are software artifacts constructed by human programmers. This has a number of relevant ramifications. First, a model of a task constructed in a cognitive architecture is runnable and produces a sequence of behaviors. These behavior sequences can be compared with the sequences produced by human users to help assess the quality of a particular model. They may also provide insight into alternate ways to perform a task—that is, they may show possible strategies that are not actually utilized by the people performing the task. This can be useful in guiding interface design as well. Another feature of many architectures is that they enable the creation of quantitative models. For instance, the model may say more than just "click on button A and then menu B," but may include the time between the two clicks as well. Models based on cognitive architectures can produce execution times, error rates, and even learning curves. This is a major strength of cognitive architectures as an approach to certain kinds of HCI problems and will be discussed in more detail in the next section.

On the other hand, cognitive architectures are large software systems, which are often considered difficult to construct and maintain. Individual models are also essentially programs, writ-

ten in the "language" of the cognitive architecture. Thus, individual modelers need to have solid programming skills.

Finally, cognitive architectures are not in wide use among HCI practitioners. Right now, they exist primarily in academic research laboratories. One of the barriers for practitioners is that learning and using most cognitive architectures is itself generally a difficult task; however, this is gradually changing, and some of the issues being addressed in this regard will be discussed in section 4. Furthermore, even if cognitive architectures are not in wide use by practitioners, this does not mean that they are irrelevant to practitioners. The next section highlights why cognitive architectures are relevant to a wide HCI audience.

RELEVANCE TO HUMAN-COMPUTER INTERACTION

For some readers, the relevance of models that produce quantitative predictions about human performance will be obvious. For others, this may be less immediately clear. Cognitive architectures (a) are relevant to usability as an engineering discipline, (b) have several HCI-relevant applications in computing systems, and (c) serve an important role in HCI as a theoretical science.

At nearly all HCI-oriented conferences, and many online resources, there are areas where corporations recruit HCI professionals. A common job title in these forums is "usability engineer." Implicit in this title is the view that usability is, at least in part, an engineering enterprise. In addition, while people with this job title are certainly involved in product design, there is a sense in which most usability engineering would not be recognized as engineering by people trained in more traditional engineering disciplines, such as electrical or aerospace engineering. In traditional engineering disciplines, design is generally guided at least in part by quantitative theory. Engineers have at their disposal hard theories of the domain in which they work, and these theories allow them to derive quantitative predictions. Consider an aerospace engineer designing a wing. Like a usability engineer, the aerospace engineer will not start with nothing; a preexisting design often provides a starting point; however, when the aerospace engineer decides to make a change in that design, there is usually quantitative guidance about how the performance of the wing will change because of the change in design. This guidance, while quantitative, is not infallible, hence the need for evaluation tools such as wind tunnels. This is similar to the usability engineer's usability test. Unlike usability testing, however, the aerospace engineer has some quantitative idea about what the outcome of the test will be, and this is not guided simply by intuition and experience, but by a mathematical theory of aerodynamics. In fact, this theory is now so advanced that few wind tunnels are built anymore. Instead, they are being replaced by computer simulations based on "computational fluid dynamics," an outcome of the application of computational techniques to complex problems in aerodynamics. Simulations have not entirely replaced wind tunnels, but the demand for wind tunnel time has clearly been affected by this development.

For the most part, the usability engineer lacks the quantitative tools available to the aerospace engineer. Every design must be subjected to its own wind tunnel (usability) test, and the engineer has little guidance about what to expect other than from intuition and experience with similar tests. While intuition and experience can certainly be valuable guides, they often fall short of more "hard" quantitative methods. Perhaps the engineer can intuit that interface "X" will allow users to complete tasks faster than with interface "Y," but how much faster? Ten percent? Twenty percent? Even small savings in execution times can add up to large financial savings for organizations when one considers the scale of the activity. The paradigm example is the telephone operators studied by Gray, John, and Atwood (1993), where even a second saved on an average call would save the telephone company millions of dollars.

Computational models based on cognitive architectures have the potential to provide detailed quantitative answers, and for more than just execution times. Error rates, transfer of knowledge, learning rates, and other kinds of performance measures are all metrics than can often be provided by architecture-based models. Even if such models are not always precisely accurate in an absolute sense, they may still be useful in a comparative sense. For example, if a usability engineer is comparing interface A with interface B and the model at his or her disposal does not accurately predict the absolute times to complete some set of benchmark tasks, it may still accurately capture the ordinal difference between the two interfaces, which may be enough.

Additionally, there are certain circumstances when usability tests are either impractical, prohibitively costly, or both. For example, access to certain populations such as physicians or astronauts may be difficult or expensive, so bringing them in for repeated usability tests may not be feasible. While developing a model of a pilot or an air traffic controller performing an expert task with specialized systems may be difficult at first, rerunning that model to assess a change made to the user interface should be much more straightforward than performing a new usability test for each iteration of the system. This is possible only with a quantitatively realistic model of the human in the loop, one that can produce things such as execution times and error rates. Computational models can, in principle, act as surrogate users in usability testing, even for special populations.

Of course, some of these measures can be obtained through other methods such as GOMS (Goals, Operators, Methods, and Selection rules) analysis (Card et al., 1983; John & Kieras, 1996) or cognitive walkthrough (Polson, Lewis, Rieman, & Wharton, 1992); however, these techniques were originally grounded in the same ideas as some prominent cognitive architectures and are essentially abstractions of the relevant architectures for particular HCI purposes. In addition, architecture-based computational models provide things that GOMS models and cognitive walkthroughs do not. First, models are executable and generative. A GOMS analysis, on the other hand, is a description of the procedural knowledge the user must have and the sequence of actions that must be performed to accomplish a specific task instance, while the equivalent computational model actually generates the behaviors, often in real time or faster. Equally importantly, computational models have the capacity to be reactive in real time. So while it may be possible to construct a GOMS model that describes the knowledge necessary and the time it will take an operator to classify a new object on an air traffic controller's screen, a paper-and-pencil GOMS model can-

not actually execute the procedure in response to the appearance of such an object. A running computational model, on the other hand, can.

Because of this property, architecture-based computational models have some other important uses beyond acting as virtual users in usability tests. One such use is in intelligent tutoring systems (ITSs). Consider the Lisp tutor (Anderson, Conrad, & Corbett, 1989). This tutoring system contained an architecture-based running computational model of the knowledge necessary to implement the relevant Lisp functions, as well as a module for assessing which pieces of this knowledge were mastered by the student. Because the model was executable, it could predict what action the student would take if the student had correct knowledge of how to solve the problem. When the student took a different action, this told the ITS that the student was missing one or more relevant pieces of knowledge. The student could then be given feedback about what knowledge was missing or incomplete, and problems that exercise this knowledge could be selected by the ITS. It is possible to generate more effective educational experiences by identifying students' knowledge, as well as the gaps in that knowledge. Problems that contain knowledge students have already mastered can be avoided, so as not to bore students with things they already know. This frees up students to concentrate on the material they have not yet mastered, resulting in improved learning (Anderson et al., 1989). While the Lisp tutor is an old research system, ITSs based on the same underlying cognitive architecture with the same essential methodology have been developed for more pressing educational needs such as algebra and geometry and are now sold commercially (see http://www.carnegielearning.com).

Another HCI-relevant application for high-fidelity cognitive models is populating simulated worlds or situations. For example, training an F-16 fighter pilot is expensive, even in a simulator, because that trainee needs to face realistic opposition. Realistic opposition consists of other trained pilots, so training one person requires taking several trained pilots away from their normal duties (e.g., flying airplanes on real missions). This is difficult and expensive. If, however, the other pilots could be realistically simulated, the trainee could face opposition that would have useful training value without having to remove already trained pilots from their duties. Many training situations such as this exist, where the only way to train someone is to involve multiple human experts who must all be taken away from their regular jobs. The need for expensive experts can potentially be eliminated (or at least reduced), however, by using architecturally based cognitive models in place of the human experts. The U.S. military has already started to experiment with just such a scenario (Jones, Laird, Nielsen, Coulter, Kenny, & Koss, 1999). Having realistic opponents is desirable in domains other than training as well, such as video games. Besides things like texture-mapped 3-D graphics, one of the features often used to sell games is network play. This enables players to engage opponents whose capabilities are more comparable to their own than typical computer-generated opponents; however, even with network play, it is not always possible for a game to find an appropriate opponent. If the computer-generated opponent were a more high-fidelity simulation of a human in terms of cognitive and perceptual-motor capabilities, then video game players would have no difficulty finding appropriate opponents without relying on network play. While this might not be the most scientifically interesting use of cognitive architectures, it seems inevitable that cognitive architectures will be used in this way.

Cognitive architectures are also theoretically important to HCI as an interdisciplinary field. Many people (including some cognitive psychologists) find terms from cognitive psychology such as "working memory" or "mental model" vague and ill defined. A harsh evaluation of explanations relying on such terms is found in Salthouse (1988, p. 3): "It is quite possible that interpretations relying on such nebulous constructs are only masquerading ignorance in what is essentially vacuous terminology." Computational cognitive architectures, on the other hand, require explicit specifications of the semantics of theoretical terms. Even if the architectures are imperfect descriptions of the human cognitive system, they are at a minimum well specified and therefore clearer in what they predict than strictly verbal theories.

A second theoretical advantage of computational theories such as cognitive architectures is that they provide a window into how the theory actually works. As theories grow in size and number of mechanisms, the interactions of those mechanisms becomes increasingly difficult to predict analytically. Computer simulations permit relatively rapid evaluations of complex mechanisms and their interactions. (For an excellent discussion of this topic, see Simon, 1996.) Another problem with theories based solely on verbal descriptions is their internal coherence can be very difficult to assess, while such assessment is much more straightforward with computational models. Verbal theories can easily hide subtle (and not so subtle) inconsistencies that make them poor scientific theories. Computational models, on the other hand, force explanations to have a high level of internal coherence; theories that are not internally consistent are typically impossible to implement on real machines.

Finally, HCI is an interdisciplinary field, and thus theories that are fundamentally interdisciplinary in nature are appropriate. Cognitive architectures are such theories, combining computational methods and knowledge from the artificial intelligence end of computer science with data and theories from cognitive psychology. While cognitive psychology and computer science are certainly not the only disciplines that participate in HCI, they are two highly visible forces in the field. Psychological theories that are manifested as executable programs should be less alien to people with a computer science background than more traditional psychological theories.

Thus, cognitive architectures are clearly relevant to HCI at a number of levels. This fact has not gone unnoticed by the HCI research community. In fact, cognitive architectures have a long history in HCI, dating back to the original work of Card, Moran, and Newell (1983).

BRIEF LOOK AT PAST SYSTEMS IN HCI

The total history of cognitive architectures and HCI would be far too long to document in a single chapter; however, it is possible to touch on some highlights. While not all of the systems described in this section qualify as complete cognitive architec-

tures, they all share intellectual history with more current architectures and influenced their development and use in HCI. Finally, many of the concepts developed in these efforts are still central parts of the ongoing research on cognitive architecture. In addition, there is a natural starting point:

The Model Human Processor (MHP) and GOMS

The Psychology of Human-Computer Interaction (Card, Moran, & Newell, 1983) is clearly a seminal work in HCI, one of the defining academic works in the early days of the field. While that work did not produce a running cognitive architecture, it was clearly in the spirit of cognitive architectures and was quite influential in the development of current cognitive architectures. Two particular pieces of that work are relevant here, the Model Human Processor (MHP) and GOMS.

The MHP represents a synthesis of the literature on cognitive psychology and human performance up to that time, and sketches the framework around which a cognitive architecture could be implemented. The MHP is a system with multiple memories and multiple processors, and many of the properties of those processors and memories are described in some detail. (See Fig. 5.1.) Card, Moran, and Newell (1983) also specified the interconnections of the processors and a number of general operating principles. In this system, there are three processors:

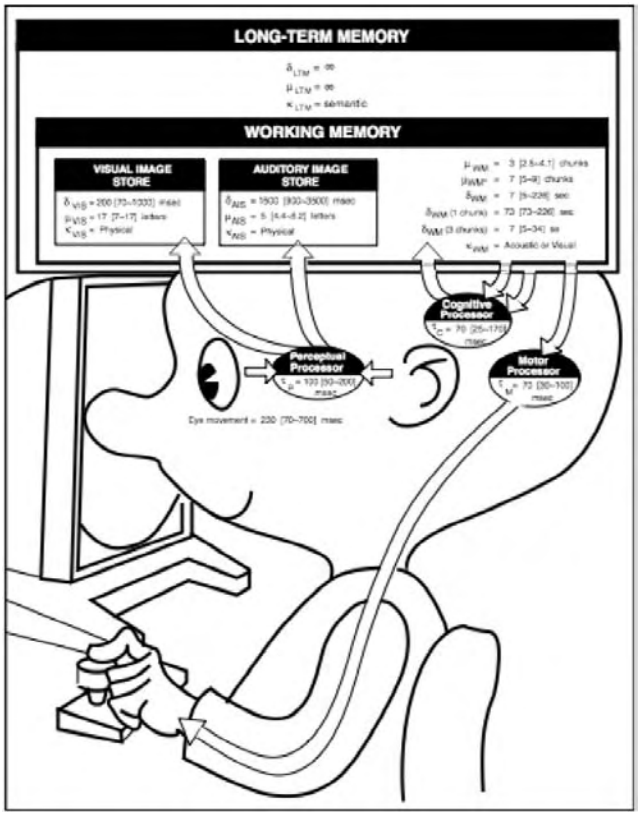


FIGURE 5.1. Model Human Processor. Based on Card, Moran, and Newell (1983).

(a) one cognitive, (b) one perceptual, and (c) one motor. In some cases, the system essentially behaves serially. For instance, in order for the system to press a key in response to the appearance of a light, the perceptual processor must detect the appearance of the light and transmit this information to the cognitive processor. The cognitive processor's job is to decide what the appropriate response should be, and then transmit that to the motor processor, which is responsible for actually executing the appropriate motor command. In this situation, the processors act serially, one after another; however, in more complex tasks such as transcription typing, all three processors often work in parallel.

Besides the specification of the timing for each processor and the connectivity of the processors, Card, Moran, and Newell named some general operating principles ranging from very general and qualitative to detailed and quantitative. For example, Principle P9, the Problem Space Principle, stated:

> The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next. (Card et al. 1983, p. 27)

This is a particularly general and somewhat vague principle. In contrast, consider Principle P5, Fitts' Law:

> The time T_{pos} to move the hand to a target of size S which lies a distance D away is given by:
>
> $$T_{pos} = I_M \log_2(D/S + .5)$$
>
> where $I_M = 100$ [70 ~ 120] ms/bit. (Card et al. 1983, p. 27).

This is a very specific principle that quantitatively describes hand movement behavior, which is highly relevant to, say, pointing with a mouse. Overall, the specification of the MHP is quite thorough, and it lays out a basis for a cognitive architecture able to do a wide variety of HCI-relevant tasks; however, Card et al. (1983) did not implement the MHP as a running cognitive architecture. This is likely for pedagogical reasons; it is not necessary to have a complete, running cognitive architecture for the general properties of that architecture to be useful for guiding HCI researchers and practitioners. At the time, computational modeling was the domain of a very specialized few in cognitive psychology.

Card et al. (1983) laid out another concept that has been highly influential throughout HCI and particularly in the community of computational modelers. This is GOMS, which stands for Goals, Operators, Methods, and Selection rules. GOMS is a framework for task analysis that describes routine cognitive skills in terms of the four listed components. Routine cognitive skills are those where the user knows what the task is and how to do the task without doing any problem solving. Text editing with a familiar editor is the prototypical case of this, but clearly, a great many tasks of interest in HCI could be classified as routine cognitive skills. Thus, the potential applicability of GOMS is quite broad. Indeed, GOMS has been applied to a variety of tasks; the website http://www.gomsmodel.org lists 143 GOMS-related papers in its bibliography.

What does a GOMS analysis provide? Essentially, a GOMS analysis of a task describes the hierarchical procedural knowledge a person must have to successfully complete that task. Based on that, and the sequence of operators that must be executed, it is possible to make quantitative predictions about the execution time for a particular task. Other analyses, such as predictions of error, functionality coverage, and learning time, are also sometimes possible. Since the original formulation presented in Card et al. (1983), a number of different forms of GOMS analysis have been developed, each with slightly different strengths and weaknesses (John & Kieras, 1996).

The core point as it relates to cognitive architectures is that GOMS analysis is originally based on a production rule analysis (Card, personal communication, 1999). Because this will come up several times, a brief introduction to production systems is warranted. Production rules are IF-THEN condition-action pairs, and a set of production rules (or simply "productions," or just "rules") and a computational engine that interprets those productions is called a "production system." In addition to productions, production systems contain some representation of the current state. This representation typically consists of a set of loosely structured data elements such as propositions or attribute-value pairs. This set is called the "working memory" or "declarative memory." Because "working memory" is also a psychological term with somewhat different meaning in that literature, "declarative memory" will be used in all further discussions.

The operation of a production system is cyclic. On each cycle, the system first goes through a pattern-matching process. The IF side of each production tests for the presence of a particular pattern in declarative memory. When the IF conditions of a production are met, the production is said to fire, and the actions specified on the THEN side are executed. The actions can be things such as pressing a button, or even some higher level abstraction of action (e.g., "turn left"). Actions also include modifying the contents of declarative memory, which usually means that a different production or productions will match on the next cycle. At this abstract and purely symbolic level, production systems are Turing complete and thus can compute anything that is computable (Newell, 1990), thus, they should be flexible enough to model the wide array of computations performed by the human cognitive system.

This is relevant to cognitive architectures because most cognitive architectures are (or contain) production systems. GOMS was actually abstracted from production rule analysis. Card et al. (1983) discovered that, for routine cognitive skills, the structure of the productions was quite similar across tasks and a more abstract representation was possible. This representation is the original GOMS formulation. Thus, translating a GOMS analysis into production rules, the language of most cognitive architectures, is generally straightforward. Similarly, for routine cognitive skills, it is often relatively simple to derive a GOMS analysis from the set of productions used to model the task. Models based on cognitive architectures can go well beyond routine cognitive skills, but this connection has certainly influenced the evolution of research on cognitive architecture and HCI. This connection has also fed back into research and development of GOMS techniques themselves, such as Natural GOMS Language (NGOMSL; Kieras, 1988). NGOMSL allows the prediction of learning time for the knowledge described in a GOMS model based on a theory of transfer of training referred to as cognitive complexity theory (CCT), which will be described in more detail in the next section.

Cognitive Complexity Theory (CCT)

When someone has learned to perform a task with a particular interface and must switch, doing the same task with a new interface, how much better off will they be than someone just learning to do the task with the new interface?—that is, how much is the knowledge gained from using the old interface "transferred" to using the new interface? This kind of question has intrigued psychologists for at least a century, and having some answers to this question has implications for training programs, user interface design, and many other areas. Cognitive complexity theory (Bovair, Kieras, & Polson, 1990; Kieras & Polson, 1985) is a psychological theory of transfer of training applied to HCI. Most relevant to the current discussion, this theory is based on production rules. The major points of CCT are as follows:

- Knowledge of the procedures that people need to execute to perform routine tasks can be represented with production rules. The relevant production rules can be generated based on a GOMS analysis of the task to be modeled.
- The complexity of a task will be reflected in the number and content of the production rules. When certain conventions are adopted about the style of those production rules, complexity is reflected almost entirely in the number of rules.
- The time it takes to execute a procedure can be predicted with a production system that interprets those rules along with a set of typical operator times, for example, the time it takes to type a three-letter command. The production interpreter used in this work was not intended to be a general cognitive architecture, but the production system framework is certainly consistent with current architectures.
- The time it takes to learn a task is a function of the number of new rules that the user must learn. "New" is clearly defined in this context. If the user already has a production, and a new task requires a rule that is similar (again, similarity is well defined based on the production rule syntax), then the rule for the new task need not be learned.
- Some predictions about errors and speedup with practice can also be gleaned from the contents of the production rules.

Obviously, this was an ambitious agenda, and there are many subtleties. For example, the notion of a "task" as the term was used in the description of CCT actually includes more than just the task at an abstract level. Consider a simple instance of a text-editing task, deleting the word "redux" from the middle of a sentence. The actual commands needed to accomplish this task could be very different in different text editors, thus, modeling the "delete word" task would require two different sets of productions, one for each editor—that is, the necessary knowledge, and thus the production rules for representing it, is actually a function both of the task from the user point of view

(e.g., "delete word") and the interface provided to accomplish the task. Transfer from one text editor to another therefore depends a great deal on the particulars of each interface. CCT thus predicts asymmetrical transfer: learning editor A after editor B should not be the same as learning editor B after editor A.

CCT models, such as a GOMS analysis, omit modeling many details of user behavior. In general, anything that falls outside the domain of procedural knowledge (how-to-do-it knowledge) is not modeled. This means that the model does not model motor actions such as keypresses, and instead has a "DoKeystroke" primitive operator. Nor do CCT models model things such as natural language comprehension, clearly a requirement in text editing. CCT models also do not include any model of the perceptual processes required by users—the model was simply given information about the state of the display and did not have to, for example, look to see if the cursor was over a particular character. This is the same scope as a typical GOMS model, though a CCT model is more formalized and quantitative than the GOMS models described by Card et al. (1983).

In spite of these limitations (or perhaps in part because these limitations allowed the researchers to concentrate on the most central aspects of the phenomena), CCT fared very well. Numerous laboratory experiments provide empirical support for many of the claims of CCT (see especially Bovair et al., 1990). The CCT framework was developed and validated in greatest detail to pre-GUI text editing, but it has also been applied to menu-based systems (Polson, Muncher, & Engelbeck, 1986) and a control panel device (Kieras & Bovair, 1986). Singley and Anderson (1989) provided a strikingly similar analysis of transfer of training, as well as supporting empirical results, lending credence to the CCT analysis. CCT was certainly one of the most prominent early successes of computational modeling in HCI.

CAPS

CAPS (Collaborative Activation-based Production System; Just & Carpenter, 1992) is a cognitive architecture designed to model individual differences in working memory (WM) capacity and the effects of working memory load. This speciality is applicable to a number of HCI situations. Certainly, some kinds of user interfaces can create excessive working memory demands, for example, phone-based interfaces. In phone-based interaction (PBI), options do not remain on a screen or in any kind of available storage; rather, users are forced to remember the options presented. This seems like a prime candidate for modeling with a system designed to capture the effects of working memory demand, and this is exactly what Huguenard, Lerch, Junker, Patz, and Kass (1997) did. Their data showed that, contrary to guideline advice and most people's intuition, restricting phone menus to only a few (three) items each does not reduce error rates. The CAPS-based model provided a clear theoretical account of this phenomenon. The model showed that short menus are not necessarily better in PBI because of two side effects of designing menu hierarchies with few options at each level. First, for the same number of total items, this increases menu depth, which creates working memory demand. Second, with fewer items at each level, each individual item has to be more general and therefore more vague, especially at the top levels of the hierarchy. This forces users to spend WM resources on disambiguating menu items when they are in a situation where WM demands outstrip supply.

Another application of CAPS that is HCI relevant is the account of postcompletion error provided by Byrne and Bovair (1997). What is a postcompletion error? Anecdotal evidence and intuition suggests that, when interacting with man-made artifacts, certain kinds of errors occur with greater frequency than others. In particular, one entire family of errors seems anecdotally common: errors that are made when some part of a task occurs after the main goal of the task has been accomplished (hence, "postcompletion"). Nearly everyone reports having made an error of this type at one time or another. Here are two prototypical examples:

1. Leaving the original on the glass of a photocopier. The main goal one generally has when using a photocopier is "get copies," and this goal is satisfied before one remove the original document. This error is less common now that many photocopiers include document feeders; the more current equivalent is leaving a document on the glass in a flatbed scanner.
2. Leaving one's bank card in an automated teller machine (ATM). Again, the main goal is something on the order of "get cash," and in many ATMs, card removal occurs after the cash is dispensed. This error was common enough in the first generation of ATMs that many ATMs are now designed in such a way that this error is now impossible to make.

Other postcompletion errors include leaving the gas cap off after filling up the car's gas tank, leaving change in vending machines, and more—most readers can probably think of several others. While numerous HCI researchers were aware of this class of error (e.g., Young, Barnard, Simon, & Whittington, 1989; Polson et al., 1992), no previous account explained why this type of error is persistent, yet not so frequent that it occurs every time. The CAPS model provides just such an account, and can serve as a useful example of the application of a cognitive architecture to an HCI problem.

Like most other production systems, CAPS contains two kinds of knowledge: (a) declarative memory and (b) productions. Declarative memory elements in CAPS also have associated with them an activation value, and elements below a threshold level of activation cannot be matched by productions' IF sides. Additionally, unlike most other production systems, the THEN side of a CAPS production may request that the activation of an element be incremented. For this to be truly useful in modeling working memory, there is a limit to the total amount of activation available across all elements. If the total activation exceeds this limit, then all elements lose some activation to bring the total back within the limit. This provides a mechanism for simulating human working memory limitations.

In Byrne and Bovair's (1997) postcompletion error model, there is a production that increments the activation of subgoals when the parent goal is active and unsatisfied. Therefore, to use the photocopier example, the "get copies" subgoal supplies activation to all the unfulfilled subgoals throughout the task; however, when the "get copies" goal is satisfied, the activation sup-

ply to the subgoals stops. Because the goal to remove the original is a subgoal of that goal, it loses its activation supply. Thus, what the model predicts is that the postcompletion subgoals are especially vulnerable to working memory load, and lower capacity individuals are more "at risk" than higher-capacity individuals. Byrne and Bovair conducted an experiment to test this prediction, and the data supported the model.

This is a good demonstration of the power of cognitive architectures. Byrne and Bovair neither designed nor implemented the CAPS architecture, but were able to use the theory to construct a model that made empirically testable predictions, and those predictions were borne out. Though it seems unlikely that CAPS will guide future HCI work (since its designers have gone in a different direction), it provides an excellent example case for a variety of reasons.

Soar

The development of Soar is generally credited to Allan Newell (especially, Newell, 1990), and Soar has been used to model a wide variety of human cognitive activity from syllogistic reasoning (Polk & Newell, 1995) to flying combat aircraft in simulated war games (Jones et al., 1999). Soar was Newell's candidate "unified theory of cognition" and was the first computational theory to be offered as such.

While Soar is a production system like CAPS, it is possible to think of Soar at a more abstract level. The guiding principle behind the design of Soar is Principle P9 from the Model Human Processor, the Problem Space Principle. Soar casts all cognitive activity as occurring in a problem space, which consists of a number of states. States are transformed through the applications of operators. Consider Soar playing a simple game like tic-tac-toe as player "X." The problem space is the set of all states of the tic-tac-toe board—not a very large space. The operators available at any given state of that space are placing an X at any of the available open spaces on the board. Obviously, this is a simplified example; the problem space and the available operators for flying an F-16 in a simulated war game are radically more complex.

Soar's operation is also cyclic, but the central cycle in Soar's operation is called a decision cycle. Essentially, on each decision cycle, Soar answers the question, "What do I do next?" Soar does this in two phases. First, all productions that match the current contents of declarative memory fire. This usually causes changes in declarative memory, so other productions may now match. Those productions are allowed to fire, and this continues until no new productions fire. At this time, the decision procedure is executed, in which Soar examines a special kind of declarative memory element, the preference. Preferences are statements about possible actions, for example, "operator o3 is better than o5 for the current operator," or "s10 rejected for supergoal state s7." Soar examines the available preferences and selects an action. Thus, each decision cycle may contain many production cycles. When modeling human performance, the convention is that each decision cycle lasts 50 ms, so productions in Soar are very low-level, encapsulating knowledge at a small grain size.

Other than the ubiquitous application of the problem space principle, Soar's most defining characteristics come from two mechanisms developed specifically in Soar: (a) universal subgoaling and (b) a general-purpose learning mechanism. Because the latter depends on the former, universal subgoaling will be described first. One of the features of Soar's decision process is that it is not guaranteed to have an unambiguous set of preferences to work with. Alternately, there may be no preferences listing an acceptable action. Perhaps the system does not know any acceptable operators for the current state, or perhaps the system lacks the knowledge of how to apply the best operator. Whatever the reason, if the decision procedure is unable to select an action, an impasse is said to occur. Rather than halting or entering some kind of failure state, Soar sets up a new state in a new problem space with the goal of resolving the impasse. For example, if multiple operators were proposed, the goal of the new problem space is to choose between the proposed operators. In the course of resolving one impasse, Soar may encounter another impasse and create another new problem space, and so on. As long as the system is provided with some fairly generic knowledge about resolving degenerate cases (e.g., if all else fails, choose randomly between the two good operators), this universal subgoaling allows Soar to continue even in cases where there is little knowledge present in the system.

Learning in Soar is a by-product of universal subgoaling. Whenever an impasse is resolved, Soar creates a new production rule. This rule summarizes the processing that went on in the substate. The resolution of an impasse makes a change to the superstate (the state in which the impasse originally occurred); this change is called a result. This result becomes the action, or THEN, side of the new production. The condition, or IF, side of the production is generated through a dependency analysis by looking at any declarative memory item matched in the course of determining this result. When Soar learns, it learns only new production rules, and it only learns as the result of resolving impasses. It is important to realize that an impasse is not equated with failure or an inability to proceed in the problem solving, but may arise simply because, for example, there are multiple good actions to take and Soar has to choose one of them. Soar impasses regularly when problem solving, and thus learning is pervasive in Soar.

Not surprisingly, Soar has been applied to a number of learning-oriented HCI situations. One of the best examples is the recent work by Altmann (Altmann & John, 1999; Altmann, 2001). Altmann collected data from an experienced programmer while she worked at understanding and updating a large computer program by examining a trace. These data included verbal protocols (e.g., thinking aloud) as well as a log of the actions taken (keystrokes and scrolling). The programmer occasionally scrolled back to find a piece of information that had previously been displayed. Altmann constructed a Soar model of her activity. This model simply attempts to gather information about its environment; it is not a complete model of the complex knowledge of an experienced programmer. The model attends to various pieces of the display, attempts to comprehend what it sees, and then issues commands. "Comprehension" in this model is manifested as an attempt to retrieve information about the object being comprehended.

When an item is attended, this creates a production that notes that the object was seen at a particular time. Because learning is pervasive, Soar creates many new rules like this, but

because of the dependency-based learning mechanism, these new productions are quite specific to the context in which the impasse originally occurred. Thus, the "index" into the model's rather extensive episodic memory consists of very specific cues, usually found on the display. Seeing a particular variable name is likely to trigger a memory for having previously seen that variable name. Importantly, this memory is generated automatically, without need for the model to deliberately set goals to remember particular items.

Altmann (2001) discussed some of the HCI ramifications for this kind of always-on episodic memory trace in terms of display clutter. While avoiding display clutter is hardly new advice, it is generally argued that it should be avoided for visual reasons (e.g., Tullis, 1983). What Altmann's model shows, however, is that clutter can also have serious implications for effective use of episodic memory. Clutter can create enormous demands for retrieval. Since more objects will generally be attended on a cluttered display, the episodic trace will be large, lowering the predictive validity for any single cue. It is unlikely that kind of analysis would have been generated if it had not been guided by a cognitive architecture that provided the omnipresent learning of Soar.

Soar has also been used to implement models of exploratory learning, somewhat in the spirit of LICAI (described later). There are two prominent models here, (a) one called IDXL (Rieman, Young, & Howes, 1996) and (b) a related model called Task-Action Learner (Howes & Young, 1996). These models both attempt to learn unfamiliar GUI interfaces. IDXL operates in the domain of using software to construct data graphs, while the Task-Action Leaner starts with even less knowledge and learns basic GUI operations such as how to open a file. For brevity, only IDXL will be described in detail.

IDXL goes through many of the same scanning processes as LICAI, but must rely on very different mechanisms for evaluation since Soar is fundamentally different from LICAI. IDXL models evaluation of various display elements as search through multiple problem spaces, one that is an internal search through Soar's internal knowledge and the other a search through the display. As items are evaluated in the search, Soar learns productions that summarize the products of each evaluation. At first, search is broad and shallow, with each item receiving a minimum of elaboration; however, that prior elaboration guides the next round of elaboration, gradually allowing IDXL to focus in on the "best" items. This model suggests a number of ways in which interface designers could thus help learners acquire a new the knowledge needed to utilize a new interface. Like the LICAI work, the IDXL work highlights the need for good labels to guide exploration. A more radical suggestion is based on one of the more subtle behavior of users and IDXL. When exploring and evaluating alternatives, long pauses often occur on particular menu items. During these long pauses, IDXL is attempting to determine the outcome of selecting the menu item being considered. Thus, one suggestion for speeding up learning of a new menu-driven GUI is to detect such pauses, and show (in some easily undoable way) what the results of selecting that item would be. For instance, if choosing that item brings up a dialog box for specifying certain options, that dialog box could be shown in some grayed-out form, and would simply vanish if the user moved off of that menu item. This would make the evaluation of the item much more certain and would be an excellent guide for novice users. This is not unlike ToolTips for toolbar icons, but on a much larger scale.

A model that does an excellent job of highlighting the power of cognitive architectures is NTD-Soar (Nelson, Lehman, & John, 1994). NTD stands for "NASA Test Director," who

> . . . is responsible for coordinating many facets of the testing and preparation of the Space Shuttle before it is launched. He must complete a checklist of launch procedures that, in its current form, consists of 3000 pages of looseleaf manuals . . . as well as graphical timetables describing the critical timing of particular launch events. To accomplish this, the NTD talks extensively with other members of launch team over a two-way radio. . . . In addition to maintaining a good understanding of the status of the entire launch, the NTD is responsible for coordinating troubleshooting attempts by managing the communication between members of the launch team who have the necessary expertise. (p. 658)

Constructing a model that is even able to perform this task at all is a significant accomplishment. Nelson, Lehman, and John (1994) were able to not only build such a model, but this model was able to produce a timeline of behavior that closely matched the timeline produced by the actual NTD being modeled—that is, the ultimate result was a quantitative model of human performance, and an accurate one at that.

It is unlikely that such an effort could have been accomplished without the use of an integrated cognitive architecture. The NTD model made extensive use of other Soar models. Nelson, Lehman, and John (1994) did not have to generate and implement theory of natural language understanding to model the communication between the NTD and others, or the NTD reading the pages in the checklist, because one had already been constructed in Soar (Lewis, 1993). They did not have to construct a model of visual attention to manage the scanning and visual search of those 3000 pages, because such a model already existed in Soar (Wiesmeyer, 1992). A great deal of knowledge engineering still took place in order to understand and model this complex task, but using an integrated architecture greatly eased the task of the modelers.

While this modeling effort was not aimed at a particular HCI problem, it is not difficult to see how it would be applicable to one. If one wanted to replace the 3000-page checklist with something like a personal digital assistant (PDA), how could the PDA be evaluated? There are very few NTDs in the world, and it is unlikely that they would be able to devote much time to participatory design or usability testing. Because an appropriate quantitative model of the NTD exists, however, it should be possible to give the model a simulated PDA and assess the impact of that change on the model's performance. Even if the model does not perfectly capture the effects of the change, it is likely that the model would identify problem areas and at least guide the developers in using any time they have with an actual NTD.

Soar has also been used as the basis for simulated agents in war games, as mentioned previously (Jones et al., 1999). This model (TacAir-Soar) participates in a virtual battle space in which humans also participate. TacAir-Soar models take on a variety of roles in this environment, from fighter pilots to helicopter crews to refueling planes. Their interactions are more complex than simple scripted agents, and they can interact with humans in the environment with English natural language. One

of the major goals of the project is to make sure that TacAir-Soar produces humanlike behavior, because this is critical to their role, which is to serve as part of training scenarios for human soldiers. In large-scale simulations with many entities, it is much cheaper to use computer agents than to have humans fill every role in the simulation. While agents (other than the ubiquitous and generally disliked paper clip) have not widely penetrated the common desktop interface, this remains an active HCI research area, and future agents in other roles could also be based on cognitive architecture rather than more engineering-oriented AI models.

While the Soar architecture is still being used for applied AI work such as TacAir-Soar, there have been essentially no new HCI-oriented Soar efforts in the last several years. Nothing, in principle, has precluded such efforts, but for the time being, Soar is primarily an example of previous achievements using cognitive architectures in HCI, as opposed to a source of new insight.

CONTEMPORARY ARCHITECTURES

Current cognitive architectures are being actively developed, updated, and applied to HCI-oriented tasks. Two of the three most prominent systems (EPIC and ACT-R) are production systems, or tather are centrally built around production systems. While all contain production rules, the level of granularity of an individual production rule varies considerably from architecture to architecture. Each one has a different history and unique focus, but they share a certain amount of intellectual history; in particular, they have all been influenced one way or another by both the MHP and each other. At some level, they may have more similarities than differences; whether this is because they borrow from one another or because the science is converging is still an open question. The third system is somewhat different from these two production system models and will be considered first.

LICAI/CoLiDeS

LICAI (Kitajima & Polson, 1997) is a primary example of an HCI-oriented cognitive architecture not based on the production system framework. With the exception of the Soar work on exploratory learning, all of the work discussed up to this point more or less assumes that the modeled users are relatively skilled with the specific interface being used; these approaches do a poor job of modeling relatively raw novices. One of the main goals of LICAI is to address this concern. The paradigm question addressed by LICAI is, "How do users explore a new interface?"

Unlike the other architectures discussed, LICAI's central control mechanisms are not based on a production system. Instead, LICAI is built around an architecture originally designed to model human discourse comprehension: construction-integration (C-I; Kintsch, 1998). Like production systems, C-I's operation is cyclic. What happens on those cycles, however, is somewhat different from what happens in a production system. Each cycle is divided into two phases: (a) construction and (b) integration (hence, the name). In the construction phase, an initial input (e.g., the contents of the current display) is fed into a weakly constrained, rule-based process that generates a network of propositions. Items in the network are linked based on their argument overlap. For example, the goal of "graph data" might be represented with the proposition (PERFORM GRAPH DATA). Any proposition containing GRAPH or DATA would thus be linked to that goal.

Once the construction phase is completed, the system is left with a linked network of propositions. What follows is the integration phase, in which activation propagates through the network in a neural, network-like fashion. Essentially, this phase is a constraint-satisfaction phase, which is used to select one of the propositions in the network as the "preferred" one. For example, the system may need to select the next action to perform while using an interface. Action representations will be added to the network during the construction phase, and an action will be selected during the integration phase. The action will be performed, and the next cycle initiated. Various C-I models have used this basic process to select things other than actions. The original C-I system used these cycles to select between different interpretations of sentences.

LICAI has three main types of cycles: (a) one to select actions, (b) one to generate goals, and (c) one to select goals. This contrasts with how most HCI tasks have been modeled in production system architectures; in such systems, the goals are usually included in the knowledge given to the system. This is not true in LICAI; in fact, the knowledge given to LICAI by the modelers is minimal. For the particular application of LICAI, which was modeling users who knew how to use a Macintosh for other tasks (e.g., word processing) and were now being asked to plot some data using a Macintosh program called CricketGraph (one group of users actually worked with Microsoft Excel), it included some very basic knowledge about the Macintosh GUI and some knowledge about graphing. Rather than supply the model with the goal hierarchy, Kitajima and Polson (1997) gave the model the same minimal instructions that were given to the subjects. One of the major jobs of the LICAI model, then, was to generate the appropriate goals as they arose while attempting to carry out the instructions.

Again, this illustrates one of the benefits of using a cognitive architecture to model HCI tasks. Kitajima and Polson (1997) did not have to develop a theory of text comprehension for LICAI to be able to comprehend the instructions given to subjects; since LICAI is based on an architecture originally designed for text comprehension, they essentially got that functionality gratis. Additionally, they did not include just any text comprehension engine, but one that makes empirically validated predictions about how people represent the text they read. Thus, the claim that the model started out with roughly the same knowledge as the users is highly credible.

The actual behavior of the model is also revealing, as it exhibits many of the same exploratory behaviors seen in the users. First, the model pursues a general strategy that can be classified as label-following (Polson et al., 1992). The model, like the users, had a strong tendency to examine anything on the screen that had a label matching or nearly matching (e.g., a near synonym) a key word in the task instructions. When the particular subtask being pursued by the model contained well-labeled

steps, the users were rapid, which was predicted by the model. While this prediction is not counter-intuitive, it is important to note that LICAI is not programmed with this strategy. This strategy naturally emerges through the normal operation of construction-integration through the linkages created by shared arguments. The perhaps less-intuitive result—modeled successfully by LICAI—is the effect of the number of screen objects. During exploration in this task, users were slower to make choices if there were more objects on the screen, but only if those items all had what were classified as "poor" labels. In the presence of "good" labels (literal match or near synonym), the number of objects on the screen did not affect decision times of either the users or LICAI.

The programmers who implemented the programs operated by the users put in several clever direct manipulation tricks. For example, to change the properties of a graph axis, one double-clicks on the axis and a dialog box specifying the properties of that axis appears. Microsoft Excel has some functionality that is most easily accessed by drag-and-drop. Franzke (1994) found that in a majority of first encounters with these types of direct manipulations, users required hints from the experimenter to be able to continue, even after two minutes of exploration. LICAI also fails at these interactions because no appropriate links are formed between any kind of task goal and these unlabeled, apparently static screen objects during the construction phase. Thus, these screen objects tend to receive little activation during the integration phase, and actions involving other objects are always selected.

Overall, LICAI does an excellent job of capturing many of the other empirical regularities in exploratory learning of a GUI interface. This is an important issue for many interfaces—particularly, any interface that is aimed at a walk-up-and-use audience. While currently common walk-up-and-use interfaces, such at ATMs, provide simple enough functionality that this is not always enormously difficult, this is not the case for more sophisticated systems, such as many information kiosks.

More recently, LICAI has been updated (and renamed to CoLiDeS, for Comprehension-based Linked model of Deliberate Search; Kitajima, Blackmon, & Polson, 2000) to handle interaction with web pages. This involves goals that are considerably less well elucidated and interfaces with a much wider range of semantic content. In order to help deal with these complexities, LICAI has been updated with a more robust attentional mechanism and a much more sophisticated notion of semantic similarity based on Latent Semantic Analysis (LSA; Landauer & Dumais, 1997). LSA is a technique for mathematically representing the meaning of words. While the details of the mathematics are complex, the underlying idea is straightforward. To perform LSA, one needs a large corpus of documents. An enormous matrix is created, showing which words appear in which contexts (documents). This matrix is then reduced by a mathematical technique called "singular value decomposition" (a cousin of factor analysis) to a smaller dimensional space of, say, 300 by 300 dimensions. Each word can then be represented by a vector in this space. The "meaning" of a passage is simply the sum of the vectors of the individual words in the passage. The similarity of two words (or two passages) can be quantified as the cosine between the vectors representing the words (or passages), with larger numbers signifying greater semantic similarity.

While LSA is certainly surrounded by a host of mathematical and theoretical issues (it should be noted that many similar techniques that differ in a variety of subtle ways have also been developed), the key result from an HCI point of view is that the technique works. Similarity measures generated by LSA generally agree very well with human judges, even for complex language understanding tasks such as grading essays. (In fact, an LSA-based system for grading essays and similar tasks has been turned into a successful commercial product; see http://www.pearsonkt.com.)

Based on the LSA-augmented CoLiDeS, Blackmon and colleagues (2002, 2003, 2005) have developed a web usability assessment and repair technique they call "Cognitive Walkthrough for the Web" (CWW). CWW is based on the idea that when users are exploring a large website, they go through a two-stage process of parsing and understanding the page, and ultimately generate descriptions of the available actions. Using those descriptions, they judge which action is most similar to their goal, and then select the link or widget that corresponds with that action. This is somewhat more complex than simply applying LSA to all of the labels on the page and comparing each one to the goal; the page is parsed into "regions," each of which is assessed, and smaller bits of text (say, link labels) are elaborated with a C-I-based process. It is those elaborations, rather than the raw link labels, that are compared via LSA to the description of the user's goal.

Based on this, CWW identifies where users with a particular goal are likely to have navigation problems and even generates problem severity estimates. These predictions have been validated in a series of experiments (Blackmon, Kitajima, & Polson, 2003; Blackmon, Kitajima, & Polson, 2005), and tools which partially automate the process are available on the web (see http://autocww.colorado.edu/). This is a significant advance in HCI, as the problem of predicting how people use semantic similarity to guide search on the web is not a trivial one and, prior to techniques like this, has generally required extensive iteration of user tests.

EPIC

With the possible exception of the NTD model, the models discussed up to this point have been primarily "pure" cognitive models—that is, the perception and action parts of the models have been handled in an indirect, abstract way. These models focus on the cognition involved, which is not surprising given the generally cognitive background of these systems; however, even the original formulation of the MHP included processors for perception and motor control. Additionally, in fact, user interfaces have also moved from having almost exclusively cognitive demands (e.g., one had to remember or problem-solve to generate command names) to relying much more heavily on perceptual-motor capabilities. This is one of the hallmarks of the GUI, the shift to visual processing and direct manipulation rather than reliance on complex composition of commands.

Providing accurate quantitative models for this kind of activity, however, requires a system with detailed models of human perceptual and motor capabilities. This is one of the major foci

and contributions of EPIC (for executive process interactive control). EPIC is the brainchild of Kieras and Meyer (see especially 1996, 1997; Kieras, Wood, & Meyer, 1997). The overall structure of the processors and memories in EPIC is shown in Fig. 5.2. This certainly bears some surface similarity to the MHP, but EPIC is substantially more detailed. EPIC was explicitly designed to pair high-fidelity models of perception and motor mechanisms with a production system. The perceptual-motor processors represent a new synthesis of the human performance literature, while the production system is the same one used in the CCT work discussed earlier.

All EPIC processors run in parallel with one another. Therefore, while the Visual Processor is recognizing an object on the screen, the Cognitive Processor can decide what word should be spoken in response to some other input, while at the same time the Manual Motor processor is pressing a key. The information flow is typical of traditional psychological models, with information coming in through the eyes and ears, and outputs coming from the mouth and hands. More specifically, what is modeled in each of the processors is primarily time course. EPIC's Visual Processor does not take raw pixels as input and compute that those pixels represent a letter "A." Instead, it determines whether the object on the screen can be seen and at what level of detail, as well as how long it will take for a representation of that object to be delivered to EPIC's declarative memory once the letter becomes available to the Visual Processor. The appearance of the letter can actually cause a number of different elements to be deposited into EPIC's declarative memory at different times. For example, information about the letter's color will be delivered before information about the letter's identity.

Similarly, on the motor side, EPIC does not simulate the computation of the torques or forces needed to produce a particular hand movement. Instead, EPIC computes the time it will take for a particular motor output to be produced after the Cognitive Processor has requested it. This is complicated by the fact that movements happen in phases. Most importantly, each movement includes a preparation phase and an execution phase. The time to prepare a movement is dependent on the number of movement features that must be prepared for each movement and the features of the last movement prepared. Features that have been prepared for the previous movement can sometimes be reused to save time. EPIC can make repeated identical movements rapidly because no feature preparation time is necessary if the movements are identical. If they are not identical, the amount of savings is a function of how different the current movement is from the previous one. After being prepared, a movement is executed. The execution time for a movement corresponds roughly to the time it takes to physically execute the movement; the execution time for aimed movements of the hands or fingers are governed by Fitts' Law. EPIC's motor processors can only prepare one movement at a time, and can

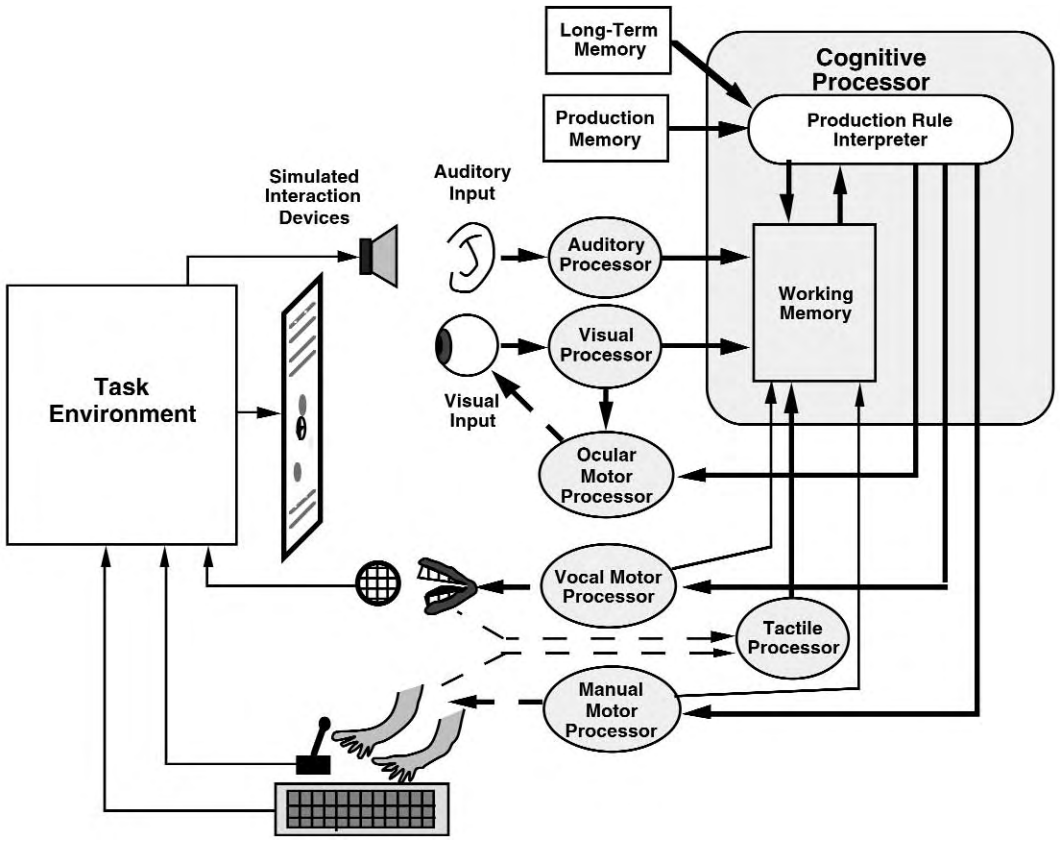


FIGURE 5.2. Overall structure of the EPIC architecture. Based on Kieras and Meyer (1996).

only execute one movement at a time, but may be preparing one movement while executing another. Thus, in some tasks, it may be possible to effectively pipeline movements in order to generate very rapid sequences of movements.

EPIC's Cognitive Processor is a production system, the same one that was used for the earlier CCT work. One highly salient feature of this system is that multiple rules can fire on a production cycle. In fact, there is no upper bound on the number of productions that can fire on a cycle. Productions in this system are at a much higher grain size than productions in Soar, which gives EPIC a parallel quality at all levels—that is, all the processors work in parallel, and EPIC's cognitive processor is itself capable of parallel processing.

This allows EPIC particular leverage in multiple-task situations. When more than one task is being performed, the tasks can execute in parallel; however, many of the perceptual-motor processors are effectively serial. People only have one set of eyes that can only be aimed at one place at a time, so if multiple tasks are ongoing and they both require the eyes, there must be something that arbitrates. In EPIC, this additional knowledge about how to manage multiple tasks is termed "executive" knowledge, and the productions that implement this knowledge execute in parallel with the productions implementing the task knowledge.

Why is all this machinery and extra knowledge necessary? Because the world of HCI is changing. The GUI forced designers and analysts to consider more seriously the perceptual-motor constraints, and the propagation of computers with user interfaces away from the desktop and into mobile phones, kiosks, automobiles, and many, many other places creates a huge demand on people's ability to multitask. Multiple-task issues have largely gone unmodeled and have been outside the theoretical scope of most psychological accounts in HCI, at least before EPIC.

While LICAI and Soar have not been adequately equipped to deal with high-performance perception and action components of many tasks, EPIC is not equipped to handle some of the issues covered by other architectures. In particular, EPIC does not include any learning mechanisms, so it would be difficult to generate EPIC models for many of the domains Soar has approached successfully. This is not a fatal shortcoming, however, as there are a wide variety of domains in which learning is not a key component and where high-fidelity modeling of perception and action, along with multiple tasking, are central.

Naturally, these are the kinds of domains to which EPIC has been applied. One of the first major applications of EPIC was to a deceptively simple dual-task paradigm known as the "psychological refractory period" (or PRP; see Meyer & Kieras, 1997). In this task, laboratory subjects are typically confronted with two choice reaction time tasks, something on the order of, "Either a red light or a green light will appear. If it's red, hit the 'L' key. If it's green, hit the 'J' key." This sounds simple, but the empirical literature is rich and shows a variety of subtle effects, for which EPIC provides the first unified account. Critically, what the EPIC models of these experiments show is that people's low-level strategies for scheduling the tasks play a large role in determining performance in this simple paradigm.

EPIC has been used to model several tasks with a more directly HCI-oriented flavor. One of those tasks is menu selection (Kieras & Meyer, 1997; Hornof & Kieras, 1997, 1999), but for brevity, a detailed description of these models will be omitted. Another application of EPIC that definitely merits mention is the model of telephone assistance operators (TAOs), data originally presented in Gray et al. (1993). When a telephone customer dials "0," a TAO is the person who answers. The TAOs modeled here sat at a "dumb terminal" style workstation and assisted customers in completing telephone calls. In particular, TAOs determine how calls should be billed, and this is done by speaking to the customer. The detailed EPIC models (Kieras et al., 1997) covered a subset of the possible billing types.

This provided a good test of EPIC because the task was performed under time pressure, and seconds—actually, milliseconds—counted in task performance. Second, this task is multimodal. The TAO must speak, type, listen, and look at a display. Third, very fine-grained performance data were available to help guide model construction. By now, it should come as no surprise to the reader that it was possible to construct an EPIC model that did an excellent job of modeling the time course of the TAO's interaction with the customer; however, this modeling effort went beyond just that and provided some insight into the knowledge engineering problem facing modelers utilizing cognitive architectures as well.

Like other production system models, EPIC provides a certain amount of freedom to the modeler in model construction. While the architecture used provides certain kinds of constraints, and these constraints are critical in doing good science and affecting the final form of the model (Howes & Young, 1997), the modeler does have some leeway in writing the production rules in the model. This is true even when the production rule model is derived from another structured representation such as a GOMS model, which was the case in the TAO model. In EPIC, it is possible to write a set of "aggressive" productions that maximize the system's ability to process things in parallel, while it is also possible to write any number of less aggressive sets representing more conservative strategies. EPIC will produce a quantitative performance prediction regardless of the strategy, but which kind of strategy should the modeler choose? There is generally no a priori basis for such a decision, and it is not clear that people can accurately self-report on such low-level decisions.

Kieras et al. (1997) generated an elegant approach to this problem, later termed "bracketing" (Kieras & Meyer, 2000). The idea is this: construct two models, one of which is the maximally aggressive version. At this end of the strategy spectrum, the models contain very little in the way of cognition. The Cognitive Processor does virtually no deliberation and spends most of its cycles simply reading off perceptual inputs and immediately generating the appropriate motor output. This represents the "super-expert" whose performance is limited almost entirely by the rate of information flow through the perceptual-motor systems. At the other end of the spectrum, a model incorporating the slowest reasonable strategy is produced. The slowest reasonable strategy is one where the basic task requirements are met, but with no strategic effort made to optimize scheduling to produce rapid performance. The idea is that observed performance should fall somewhere in between these two extremes, hence, the data should be bracketed by the two versions of the model. Different users will tend to perform at different ends of this range for different tasks, so this is an excellent way to ac-

commodate some of the individual differences that are always observed in real users.

What was discovered by employing this bracketing procedure to the TAO models was surprising. Despite the fact that the TAOs were under considerable time pressure and were extremely well practiced experts, their performance rarely approached the fastest possible model. In fact, their performance most closely matched a version of the model termed the "hierarchical motor-parallel model." In this version of the model, eye, hand, and vocal movements are executed in parallel with one another when possible; furthermore, the motor processor is used somewhat aggressively, preparing the next movement while the current movement was in progress. The primary place where EPIC could be faster but the data indicated the TAOs were not was in the description of the task knowledge. It is possible to represent the knowledge for this task as one single, flat GOMS method with no use of subgoals. On the other hand, the EPIC productions could represent the full subgoal structure or a more traditional GOMS model. Retaining the hierarchical representation—thus incurring time costs for goal management—provided the best fit to the TAOs performance. This provides solid evidence for the psychological reality of the hierarchical control structure inherent in GOMS analysis, since even well-practiced experts in fairly regular domains do not abandon it for some kind of faster knowledge structure.

The final EPIC model that will be considered is the model of the task first presented in Ballas, Heitmeyer, and Perez (1992). Again, this model first appeared in Kieras and Meyer (1997), but a richer version of the model is described in more detail later, in Kieras, Meyer, and Ballas (2001). The display used is a split screen. On the right half of the display, the user was confronted with a manual tracking task, which was performed using a joystick. The left half of the display was a tactical task in which the user had to classify objects as hostile or neutral based on their behavior. There were two versions of the interface to the tactical task, one a command-line style interface using a keypad and one a direct-manipulation-style interface using a touch screen. The performance measure of interest in this task is the time taken to respond to events (such as the appearance of a new object or a change in state of an object) on the tactical display.

This is again a task well suited to EPIC because the perceptual-motor demands are extensive. This is not, however, what makes this task so interesting. What is most interesting is the human performance data: in some cases, the keypad interface was faster than the touch screen interface, and in many cases, the two yielded almost identical performance, while in some other cases, the touch screen was faster. Thus, general claims about the superiority of GUIs do not apply to this case and a more precise and detailed account is necessary.

EPIC provides just the tools necessary to do this. Two models were constructed for each interface, again using the bracketing approach. The results were revealing. In fact, the fastest possible models showed no performance advantage for either interface. The apparent direct-manipulation advantage of the touch screen for initial target selection was almost perfectly offset by some type-ahead advantages for the keypad. The reason for the inconsistent results is that the users generally did not operate at the speed of the fastest possible model; they tended to work somewhere in between the brackets for both interfaces. However, they tended to work more toward the upper (slowest-reasonable) bracket for the touch screen interface. This suggests an advantage for the keypad interface, but the caveat is that the slowest reasonable performance bound for the touch screen was faster than the slowest possible for the keypad. Thus, any strategy changes made by users in the course of doing the task, perhaps as a dynamic response to changes in workload, could affect which interface would be superior at any particular point in the task. Thus, results about which interface is faster are likely to be inconsistent—which is exactly what was found.

This kind of analysis would be impossible to conduct without a clear, quantitative human performance model. Constructing and applying such a model also suggested an alternative interface that would almost certainly be faster than either, which is one using a speech-driven interface. One of the major performance bottlenecks in the task was the hands, and so in this case, voice-based interaction should be faster. Again, this could only be clearly identified with the kind of precise quantitative modeling enabled by something like the EPIC architecture.

ACT-R 5.0

ACT-R 5.0 (Anderson et al., 2004) represents another approach to a fully unified cognitive architecture, combining a very broad model of cognition with rich perceptual-motor capabilities. ACT-R 5.0 is the most recent iteration of the ACT-R cognitive architecture (introduced in Anderson, 1993). ACT-R has a long history within cognitive psychology, as various versions of the theory have been developed over the years. In general, the ACT family of theories has been concerned with modeling the results of psychology experiments, and this is certainly true of the current incarnation, ACT-R. Anderson and Lebiere (1998) show some of this range, covering areas as diverse as list memory (chapter 7), choice (chapter 8), and scientific reasoning (chapter 11).

ACT-R is, like EPIC and Soar, a production system with activity centered on the production cycle, which is also set at 50 ms in duration; however, there are many differences between ACT-R and the other architectures. First, ACT-R can only fire one production rule per cycle. When multiple production rules match on a cycle, an arbitration procedure called conflict resolution comes into play. Second, ACT-R has a well-developed theory of declarative memory. Unlike EPIC and Soar, declarative memory elements in ACT-R are not simply symbols. Each declarative element in ACT-R also has an activation value associated with it, which determines whether and how rapidly it may be accessed. Third, ACT-R contains learning mechanisms, but is not a pervasive learning system in the same sense as Soar. These mechanisms are based on a "rational analysis" (Anderson, 1990) of the information needs of an adaptive cognitive system.

For example of rational analysis, consider conflict resolution. Each production in ACT-R has associated with it several numeric parameters, including numbers which represent the probability that if the production fires, the goal will be reached and the cost, in time, that will be incurred if the production fires. These values are combined according to a formula that trades off probability of success vs. cost and produces an "expected gain" for each production. The matching production with the highest expected gain is the one that gets to fire when conflict resolu-

tion is invoked. The expected gain values are noisy, so the system's behavior is somewhat stochastic, and the probability and cost values are learned over time so that ACT-R can adapt to changing environments.

Similarly, the activation of elements in declarative memory is based on a Bayesian analysis of the probability that a declarative memory element will be needed at a particular time. This is a function of the general utility of that element, reflected in what is termed its "base-level" activation, and that element's association with the current context. The more frequently and recently an element has been accessed, the higher its base-level activation will be, and thus the easier it is to retrieve. This value changes over time according to the frequency and recency of use, thus this value is learned. These kinds of mechanisms have helped enable ACT-R to successfully model a wide range of cognitive phenomena.

The current version of the theory, 5.0, incorporates several important changes over previous versions of the theory. First, the architecture is now a fully modular architecture, with fully separate and independent modules for different kinds of information processing (such as declarative memory and vision). Second, ACT-R 5.0 no longer contains a "goal stack" for automated goal management; goals are now simply items in declarative memory and subject to processes like learning and decay. Third, the new modular architecture allows processing in ACT-R to be mapped onto the human brain, with different modules identified with different brain regions. Finally, a production rule learning mechanism has been fully incorporated, so learning is now more pervasive in ACT-R. The 5.0 version of the theory subsumes all previous versions of the theory including ACT-R/PM, which is now considered obsolete.

In ACT-R 5.0, the basic production system is augmented with four EPIC-like peripheral modules, as well as a goal module and a declarative memory module, as depicted in Fig. 5.3. Like EPIC, all of these modules run in parallel with one another, giving ACT-R the ability to overlap processing. The peripheral modules come from a variety of sources. ACT-R's Vision Module is based on the ACT-R Visual Interface described in Anderson, Matessa, and Lebiere (1997). This is a feature-based attentional visual system, but does not explicitly model eye movements. Recently, the Vision Module has been extended to include an eye-movement model (Salvucci, 2001a) as well. The Motor Module is nearly identical to the Manual Motor Processor in EPIC and is based directly on the specification found in Kieras and Meyer (1996), and the Speech Module is similarly derived from EPIC. The Audition Module is a hybrid of the auditory system found in EPIC and the attentional system in ACT-R's Vision Module.

One other important property of ACT-R is that it is possible to have models interact with the same software as the human users being modeled. The software development conditions are fairly restrictive, but if these conditions are met, then both the user and the model are forced to use the same software. This reduces the number of degrees of freedom available to the modeler in that it becomes impossible to force any unpleasant modeling details into the model of the user interface, because there is no model of the user interface. More will be said about this issue in the next section.

ACT-R has been applied to numerous HCI domains. The first example comes from the dissertation work of Ehret (1999).

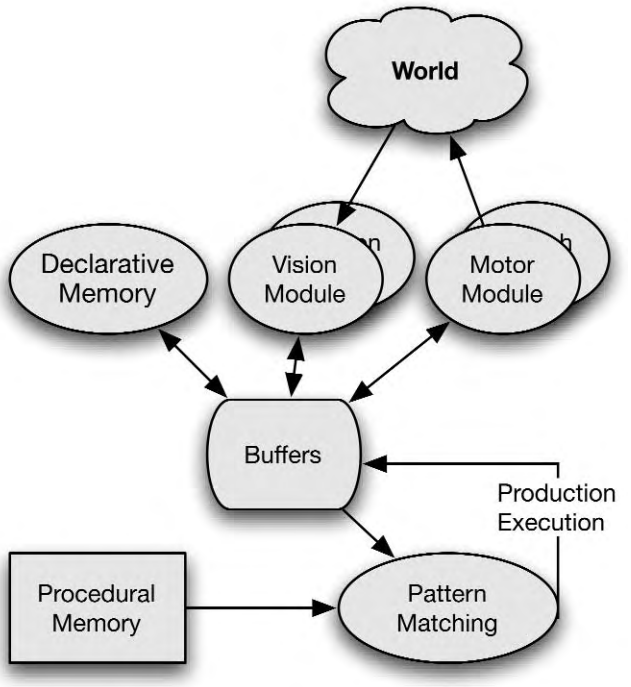


FIGURE 5.3. Overall structure of the ACT-R 5.0 architecture.

Among other things, Ehret developed an ACT-R/PM model of a fairly simple, but subtle, experiment. In that experiment, subjects were shown a target color and asked to click on a button that would yield that color. The buttons themselves had four types: (a) blank, (b) arbitrary icon, (c) text label, and (d) color. In the color condition, the task was simple: users simply found the color that matched the target and then clicked the button. In the text label condition, the task was only slightly more difficult: users could read the labels on the buttons and select the correct one because the description matched the color. In the arbitrary icon condition, more or less random pictures appeared on each icon (e.g., a mailbox). Users had to either memorize the picture-to-color mapping, which they had to discover by trial and error, or memorize the location of each color, since the buttons did not change their functions over time. In the hardest condition, the blank condition, users simply had to memorize the mapping between button location and color, which they had to discover through trial and error.

Clearly, the conditions will produce different average response times, and what Ehret (1999) found is that they also produced somewhat different learning curves over time. Ehret added an additional manipulation as well: after users performed the task for some time, all labeling was removed. Not surprisingly, the amount of disruption was different in the different conditions, reflecting the amount of incidental location learning that went on as subjects performed the task. The ACT-R model that Ehret constructed did an excellent job of explaining the results. This model represented the screen with the built-in visual mechanisms from ACT-R and learned the mappings between color and location via ACT-R's standard associative learning mechanisms. The initial difference between the various conditions was repro-

duced, as were the four learning curves. The model also suffered disruptions similar to those suffered by the human users when the labels were removed. This model is an excellent demonstration of the power of ACT-R, exercising both the perceptual-motor capabilities of the system as well as the graded learning in ACT-R's rational-analysis driven mechanisms.

Salvucci (2001b) described an ACT-R model that tests ACT-R's ability to handle multimodal, high-performance situations in a very compelling task: this model drives an automobile driving simulator. This is not a robotics project; the ACT-R model does not actually turn the steering wheel or manipulate the pedals; rather, it communicates with the automobile simulation software. The model's primary job is to maintain lane position as the car drives down the road. Salvucci (2001b) added an additional task that makes it particularly interesting: the model dials telephone numbers on a variety of mobile phone interfaces. Two factors were crossed: (a) whether the telephone was dialed manually via keypad vs. dialed by voice, and (b) whether the full telephone number needed to be dialed vs. a shortened "speed dial" system. The model was also validated by comparison with data from human users.

Both the model and the human users showed that dialing while not driving is faster than dialing while driving, and that steering performance can be disrupted by telephone dialing. Not surprisingly, the most disruptive interface was the "full-manual" interface, where the full phone numbers were dialed on a keypad. This is due largely to the fact that dialing with the keypad requires visual guidance, causing the model (and the users) to take their eyes off the road. There was very little disruption associated with the voice interfaces, regardless of whether full numbers or speed-dial was used.

This is a powerful illustration of the value of cognitive architectures for a number of reasons. First, the basic driving model could simply be reused for this task; it did not have to be reimplemented. Second, the model provides an excellent quantitative fit to the human data. Third, this is an excellent example of a situation where testing human users can be difficult. Testing human drivers with interfaces that degrade driving performance is dangerous, so simulators are generally used for this kind of evaluation; however, maintaining a driving simulator requires a great deal of space and is quite expensive. If someone wanted to test another variant of the telephone interface, it would be much faster and cheaper to give that interface to Salvucci's (2001b) model than it would be to recruit and test human drivers.

One of the long-standing criticisms of cognitive architectures in HCI is that they are theoretical tools designed to support research and are too difficult for the average practitioner to use in a more applied context. This criticism certainly holds some truth (though how much may be open to debate); however, some researchers are actively trying to address it. One of the most interesting developments in the cognitive architecture arena in the past few years has been the work of John and colleagues (2004) on something they term "CogTool" (see http://www.cogtool.org). CogTool allows interface designers to mock up an interface in HTML, demonstrate the tasks to be performed in a GUI environment, and have an ACT-R based cognitive model of the tasks automatically synthesized. Alternative interfaces can be mocked up, and predicted performance compared, giving the designer important information about how changes in the interface may impact user performance. This is not limited strictly to prediction of time of execution, either, as the full model output—a complete trace of the predicted user behavior—is also available for inspection.

This is an important advance, as it makes the power of complete and detailed computational cognitive modeling available to people with no experience with cognitive architectures. It also has important ramifications for the methodology of model building, since models generated by CogTool are not hand-tweaked in an attempt to optimize fit to data; in fact, the intent is to make predictions about nonexistent data. Some validation, however, has been performed, and early explorations with CogTool have found the time predictions to be quite accurate. As exciting a development as this is, it is important to note that there are clearly limitations. The language of available actions is based on the keystroke-level model (KLM) of Card, Moran, and Newell (1983) and is thus somewhat limited. For example, it does not provide a model for any kind of cognition more complex than very simple decisions about where to point. For example, if one wanted to model use of an ATM, CogTool would not model the process of the user trying to retrieve a security code from memory. (However, if the user was highly practiced such that this retrieval is very rapid and always successful, then CogTool could be appropriate.) Efforts are underway to make automatic translation of slightly more complex GOMS-style models into ACT-R models possible (St. Amant, Freed, & Ritter, 2005), but that still would not address the more complex cognition required by many interfaces. Nonetheless, CogTool represents a significant advance in making architecture-based modeling accessible to those who are not experts in the field.

The last application of ACT-R in HCI to be covered in detail is based on Pirolli and Card's (1999) theory of *information foraging*. This theory is based on the idea that people forage for information in much the same way that animals forage for food. The literature on the latter type of foraging is quite rich, including equations that describe when foragers will give up on a particular patch of information and move on to the next one. Foraging for food is often guided by proximal cues (e.g., scent) that indicate the presence of distal content (e.g., food); analogously, information foragers may use proximal cues (e.g., visible link labels) to guide their search for information that they cannot presently access. In the spirit of the analogy, Pirolli and Card term these proximal cues "information scent." That such a theory should be applicable to the web should be fairly obvious.

This theory has spawned a number of models based on modified versions of ACT-R. The original model from Pirolli and Card (1999) was called ACT-IF (for "information foraging"). Later versions were SNIF-ACT 1.0 (Card et al., 2001) and SNIF-ACT 2.0 (Pirolli & Fu, 2003), where SNIF stands for "scent-based navigation and information foraging." These models use a modified form of ACT-R's spreading activation to compute information scent, which drives the model's choices as it browses through web pages. SNIF-ACT 1.0 and later also uses an altered version of ACT-R's conflict resolution mechanism to determine when to abandon the current website (or information "patch")

and move on to a new site. SNIF-ACT 2.0 adds a further improvement, which is the use of a new measure to compute the strengths of association (and therefore spreading activation) between words. This new measure, termed "pointwise mutual information" (PMI), is much easier to compute than LSA, but generally produces similar results. SNIF-ACT 2.0 was used to predict link selection frequency in a large ($N = 244$) Web behavior study with impressive results, accounting for 72% of the selection variance on one site and 90% of the variance on a second site.

Obviously, this work is clearly related to CWW. In fact, the foraging work has also led to a tool to help automate the process of web usability assessment, this one named "Bloodhound" (Chi et al., 2003). Bloodhound takes a site and a goal description and uses scent metrics combined with Monte Carlo methods to compute usage patterns and identify navigation problems. The two projects clearly have similar goals and similar outputs. What is less clear is the extent to which the differences in the underlying cognitive architecture and differences in other aspects of the techniques (such as LSA vs. PMI) actually lead to different predictions about user behavior. As of this writing, the two systems have never both been applied to the same website. However appealing such a "bake-off" might be, it is important not to lose sight of the advance even the possibility of such a comparison represents: it is only recently that the field has had psychologically informed and quantitative predictions about something as complex as Web behavior available for potential comparison. Thus, this is a concrete example of cognitive architectures paying off in HCI.

Other published and ongoing work applying ACT-R to HCI-oriented problems is available (e.g., Taatgen & Lee, 2003; Peebles & Cheng, 2003; Byrne, 2001; Schoelles & Gray, 2000), but space considerations prohibit a more exhaustive review. The vitality of the research effort suggests that ACT-R's combination of perceptual-motor modules and a strong theory of cognition will continue to pay dividends as an HCI research tool.

In fact, this is not limited to ACT-R; overall, cognitive architectures are an exciting and active area of HCI research. The systems described here all take slightly different approaches and focus on slightly different aspects of various HCI problems, but there is clearly a great deal of cross-pollination. Lessons learned and advancements made in one architecture often affect other systems, for example, the development of EPIC's peripheral systems clearly impacted ACT-R.

Comparisons

An exhaustive and detailed comparison of the major cognitive architectures is beyond the scope of this chapter; however, an excellent comparison that includes a number of other architectures can be found in Pew and Mavor (1998). Certainly, the three production systems are related: (a) Soar, (b) EPIC, and (c) ACT-R. A major difference between them is their original focus; they were originally developed to model slightly different aspects of human cognition. As they develop, however, there appears to be more convergence than divergence. This is generally taken as a good sign that the science is cumulating. Still, there are differences, and certainly between the production systems and LICAI/CoLiDeS. Many of the relevant comparisons are summarized in Table 5.1.

Most of the entries on this table have been previously discussed, with the exception of the last two. Support for learning will be discussed in the next section. The presence and size of the user community has not been discussed, as it is not clear what role (if any) such a community plays in the veridicality of the predictions made by the system; however, it may be relevant to researchers for other reasons, particularly those trying to learn the system.

In addition, many of the values on this table are likely to change in the future. For example, the 6.0 version of ACT-R will be available by the time this chapter appears (though there are few major theoretical changes planned), and planning for a major revision for version 7.0 is already underway.

It is difficult to classify the value an architecture has on a particular attribute as an advantage or a disadvantage, because what constitutes an advantage for modeling one phenomenon may be a disadvantage for modeling others. For example, consider learning in Soar. Certainly, when attempting to model the improvement of users over time with a particular interface, Soar's learning mechanism is critical; however, there are many applications for which modeling learning is not critical, and Soar's pervasive learning feature occasionally causes undesired side effects that can make model construction more difficult.

TABLE 5.1. Architecture Feature Comparison

	LICAI/CoLiDeS	EPIC	ACT-R 5.0
Original focus	Text comprehension	Multiple-task performance	Memory and problem-solving
Basic cycle	Construction- integration	Production cycle (parallel)	Production cycle (serial)
Symbolic or activation-based?	Both	Symbolic	Both
Architectural goal management	Special cycle types, supports Vague goals	None	None
Detailed perceptual-motor systems	No	Yes	Yes
Learning mechanisms	No	No	Yes
Large, integrated models	No	No	Soon
Extensive natural language	Yes	No	No
Support for learning system	None	None	Extensive tutorial materials, summer school
User community*	None	None	Growing, primarily psychology

*Outside of the researchers who have developed the system.

THE FUTURE OF COGNITIVE ARCHITECTURES IN HCI

Beyond the incremental development and application of architectures such as EPIC and ACT-R, what will the future hold for cognitive architectures in HCI? What challenges are faced, and what is the ultimate promise? Indeed, a number of important limitations for cognitive architectures currently exist. There are questions they cannot yet address, and it is hard to see how they even would address some questions. Other limitations are more pragmatic than in principle, but these are relevant as well.

First, there is a wide array of HCI problems that are simply outside the scope of current cognitive architectures. Right now, these architectures focus on cognition and performance, but not on other aspects of HCI, such as user preference, boredom, aesthetics, fun, and so on. Another important challenge, though one that might be overcome, is that these architectures generally have not been applied to social situations, such as those encountered in groupware or online communities (Olson & Olson, this volume; Preece, this volume). In principle, it is possible to implement a model of social interaction in a cognitive architecture; however, the knowledge engineering problem here would certainly be a difficult one. How does one characterize and implement knowledge about social situations with enough precision to be implemented in a production system? It may ultimately be possible to do so, but it is unlikely that this will happen anytime soon.

One problem that will never entirely be resolved, no matter how diligent the modelers are, is the knowledge engineering problem. Every model constructed using a cognitive architecture still needs knowledge about how to use the interface and what the tasks are. By integrating across models, the knowledge engineering demands when entering a new domain may be reduced (the NTD is a good example), but they will never be eliminated. This requirement will persist even if an architecture was to contain a perfect a theory of human learning—and there is still considerable work to be done to meet that goal.

Another barrier to the more widespread use of cognitive architectures in HCI is that the architectures themselves are large and complex pieces of software, and (ironically) little work has been done to make them usable or approachable for novices. For example, "EPIC is not packaged in a 'user-friendly' manner; full-fledged Lisp programming expertise is required to use the simulation package, and there is no introductory tutorial or user's manual" (Kieras & Meyer, 1997, p. 399). The situation is slightly better for ACT-R: tutorial materials, documentation, and examples for ACT-R are available, and most years there is a two-week "summer school" for those interested in learning ACT-R (see http://act.psy.cmu.edu).

Another limiting factor is implementation. In order for a cognitive architecture to accurately model interaction with an interface, it must be able to communicate with that interface. Because most user interfaces are "closed" pieces software with no built-in support for supplying a cognitive model with the information it needs for perception (e.g., what is on the screen where) or accepting input from a model, this creates a technical problem. Somehow, the interface and the model must be connected. An excellent summary of this problem can be found in Ritter, Baxter, Jones, and Young (2000). A number of different approaches have been taken. In general, the EPIC solution to this problem has been to reimplement the interface to be modeled in Lisp (more recent versions of EPIC require C++), so the model and the interface can communicate via direction function calls. The ACT-R solution is not entirely dissimilar. In general, ACT-R has only been applied to relatively new experiments or interfaces that were initially implemented in Lisp, and thus ACT-R and the interface can communicate via function calls. In order to facilitate the construction of models and reduce the modeler's degrees of freedom in implementing a custom interface strictly for use by a model, ACT-R does provide some abilities for automatically managing this communication when the interface is built with the native GUI builder for Macintosh Common Lisp under MacOS and Allegro Common Lisp under Windows. If the interface is implemented this way, both human users and the models can interact with the same interface.

The most intriguing development along this line, however, is recent work by St. Amant and colleagues (St. Amant & Riedl, 2001; St. Amant, Horton, & Ritter, in press). They have implemented a system called SegMan, which directly parses the screen bitmap on Windows systems—that is, given a Windows display—any Windows display—SegMan can parse it and identify things such as text, scroll bars, GUI widgets, and the like. It also has facilities for simulating mouse and keyboard events. This is an intriguing development, because in principle, it should be possible to connect this to an architecture such as EPIC or ACT-R, which would enable the architecture to potentially work with any Windows application in its native form.

While a number of technical details would have to be worked out, this has the potential of fulfilling one of the visions held by many in the cognitive architecture community: a high-fidelity "virtual user" that could potentially use any application or even combination of applications. Besides providing a wide array of new domains to researchers, this could be of real interest to practitioners as well because this opens the door for at least some degree of automated usability testing. This idea is not a new one (e.g., Byrne, Wood, Sukaviriya, Foley, & Kieras, 1994; St. Amant, 2000), but technical and scientific issues have precluded its adoption on even a limited scale. This would not eliminate the need for human usability testing (see Ritter & Young, 2001, for a clear discussion of this point) for some of the reasons listed above, but it could significantly change usability engineering practice in the long run.

The architectures themselves will continue to be updated and applied to more tasks and interfaces. As mentioned, a new version of ACT-R (version 6.0) is currently under development, and this new version has definitely been influenced by issues raised by HCI concerns. New applications of EPIC result in new mechanisms (e.g., similarity-based decay in verbal working memory storage; Kieras, Meyer, Mueller, & Seymour, 1999) and new movement styles (e.g., click-and-point, Hornof & Kieras, 1999). Applications such as the World Wide Web are likely to drive these models into more semantically rich domain areas, and tasks that involve greater amounts of problem solving are also likely candidates for future modeling.

Despite their limitations, this is a particularly exciting time to be involved in research on cognitive architectures in HCI. There is a good synergy between the two areas, as cognitive architec-

tures are certainly useful to HCI, so HCI is also useful for cognitive architectures. HCI is a complex and rich yet still tractable domain, which makes it an ideal candidate for testing cognitive architectures. HCI tasks are more realistic and require more integrated capabilities than typical cognitive psychology laboratory experiments, and thus cognitive architectures are the best theoretical tools available from psychology. Theories like EPIC-Soar (Chong & Laird, 1997) and ACT-R are the first complete psychological models that go from perception to cognition to action in detail. This is a significant advance and holds a great deal of promise.

The need for truly quantitative engineering models will only grow as user interfaces propagate into more and more places and more and more tasks. Cognitive architectures, which already have a firmly established niche in HCI, seem the most promising road toward such models. Thus, as the architectures expand their range of application and their fidelity to the humans they are attempting to model, this niche is likely to expand. HCI is an excellent domain for testing cognitive architectures as well, so this has been, and will continue to be, a fruitful two-way street.

References

Altmann, E. M. (2001). Near-term memory in programming: A simulation-based analysis. *International Journal of Human-Computer Studies, 54*(2), 189–210.

Altmann, E. M., & John, B. E. (1999). Episodic indexing: A model of memory for attention events. *Cognitive Science, 23*(2), 117–156.

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*, 1036–1060.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science, 13*(4), 467–505.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction, 12*(4), 439–462.

Ballas, J. A., Heitmeyer, C. L., & Perez, M. A. (1992). Evaluating two aspects of direct manipulation in advanced cockpits. *Proceedings of ACM CHI 92 Conference on Human Factors in Computing Systems* (pp. 127–134). New York: ACM.

Blackmon, M. H., Kitajima, M., & Polson, P. G. (2003). Repairing usability problems identified by the Cognitive Walkthrough for the Web. In *Human factors in computing systems: Proceedings of CHI 2003* (pp. 497–504). New York: ACM.

Blackmon, M. H., Kitajima, M., & Polson, P. G. (2005). Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. In *Human factors in computing systems: Proceedings of CHI 2005* (pp. 31–40). New York: ACM.

Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C. (2002). Cognitive walkthrough for the web. In *Human factors in computing systems: Proceedings of CHI 2002* (pp. 463–470). New York: ACM.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*(1), 1–48.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies, 55*, 41–84.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science, 21*(1), 31–61.

Byrne, M. D., Wood, S. D., Sukaviriya, P. N., Foley, J. D., & Kieras, D. E. (1994). Automating interface evaluation. *ACM CHI'94 Conference on Human Factors in Computing Systems* (pp. 232–237). New York: ACM Press.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Card, S. K., Pirolli, P., Van Der Wege, M., Morrison, J. B., Reeder, R. W., Schraedley, P. K., et al. (2001). Information scent as a driver of Web behavior graphs: Results of a protocol analysis method for Web usability. In *Human factors in computing systems: Proceedings of CHI 2001* (pp. 498–505). New York: ACM.

Chi, E. H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., et al. (2003). The Bloodhound project: Automating discovery of web usability issues using the Infoscent simulator. In *Human factors in computing systems: Proceedings of CHI 2003* (pp. 505–512). New York: ACM.

Chong, R. S., & Laird, J. E. (1997). Identifying dual-task executive process knowledge using EPIC-Soar. In M. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 107–112). Hillsdale, NJ: Erlbaum.

Ehret, B. D. (1999). Learning where to look: The acquisition of location knowledge in display-based interaction. Unpublished doctoral dissertation, George Mason University, Fairfax, VA.

Franzke, M. (1994). Exploration, acquisition, and retention of skill with display-based systems. Unpublished doctoral dissertation, University of Colorado, Boulder.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction, 8*(3), 237–309.

Gray, W. D., Young, R. M., & Kirschenbaum, S. S. (1997). Introduction to this special issue on cognitive architectures and human-computer interaction. *Human-Computer Interaction, 12*, 301–309.

Hornof, A., & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems* (pp. 107–114). New York: ACM.

Hornof, A., & Kieras, D. (1999). Cognitive modeling demonstrates how people use anticipated location knowledge of menu items. *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems* (pp. 410–417). New York: ACM.

Howes, A., & Young, R. M. (1996). Learning consistent, interactive, and meaningful task-action mappings: A computational model. *Cognitive Science, 20*(3), 301–356.

Howes, A., & Young, R. M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human-Computer Interaction, 12*(4), 311–343.

Huguenard, B. R., Lerch, F. J., Junker, B. W., Patz, R. J., & Kass, R. E. (1997). Working memory failure in phone-based interaction. *ACM Transactions on Computer-Human Interaction, 4*, 67–102.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*, 320–351.

John, B. E., Prevas, K., Savucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. In *Human factors in*

computing systems: Proceedings of CHI 2004 (pp. 455–462). New York: ACM.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine, 20*(1), 27–41.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review, 99*(1), 122–149.

Kieras, D. E., & Bovair, S. (1986). The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory & Language, 25*(5), 507–524.

Kieras, D. E., & Meyer, D. E. (1996). The EPIC architecture: Principles of operation. Retrieved July 4, 1996, from ftp://ftp.eecs.umich .edu/people/kieras/EPICarch.ps.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*(4), 391–438.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis in the application of predictive models of human performance. In J. M. Schraagen & S. F. Chipman (Eds.), *Cognitive task analysis* (pp. 237–260). Mahwah, NJ: Erlbaum.

Kieras, D. E., Meyer, D. E., Mueller, S., & Seymour, T. (1999). Insights into working memory from the perspective of the EPIC architecture for modeling skilled perceptual-motor and cognitive human performance. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 183–223). New York: Cambridge University Press.

Kieras, D. E., Meyer, D. E., & Ballas, J. A. (2001). Towards demystification of direct manipulation: cognitive modeling charts the gulf of execution, *Proceedings of ACM CHI 01 Conference on Human Factors in Computing Systems* (pp. 128–135). New York: ACM.

Kieras, D., & Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies, 22*(4), 365–394.

Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for multimodal high-performance human-computer interaction task. *Transactions on Computer-Human Interaction, 4*(3), 230–275.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* New York: Cambridge University Press.

Kitajima, M., & Polson, P. G. (1997). A comprehension-based model of exploration. *Human-Computer Interaction, 12*(4), 345–389.

Kitajima, M., Blackmon, M. H., & Polson, P. G. (2000). A comprehension-based model of web navigation and its application to web usability analysis. In S. McDonald & Y. Waern & G. Cockton (Eds.), *People and Computers XIV-Usability or Else!* (Proceedings of HCI 2000) (pp. 357–373). New York: Springer.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211–240.

Lewis, R. L. (1993). An architecturally-based theory of human sentence comprehension. Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: I. Basic mechanisms. *Psychological Review, 104*(1), 3–65.

Nelson, G., Lehman, J. F., & John, B. E. (1994). Integrating cognitive capabilities in a real-time task. In A. Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 353–358). Hillsdale, NJ: Erlbaum.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual information processing* (pp. 283–308). New York: Academic Press.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 279–293). Cambridge, MA: MIT Press.

Peebles, D., & Cheng, P. C. H. (2003). Modeling the effect of task and graphical representation on response latency in a graph reading task. *Human Factors, 45*, 28–45.

Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling human and organizational behavior: Application to military simulations.* Washington, DC: National Academy Press.

Pirolli, P., & Card, S. (1999). Information foraging. *Psychological Review, 106*(4), 643–675.

Pirolli, P., & Fu, W. (2003). SNIF-ACT: A model of information foraging on the world wide web. In P. Brusilovsky, A. Corbett, & F. de Rosis (Eds.), *Proceedings of the ninth international conference on user modeling* (pp. 45–54). Johnstown, PA: Springer-Verlag.

Polk, T. A., & Newell, A. (1995). Deduction as verbal reasoning. *Psychological Review, 102*(3), 533–566.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies, 36*(5), 741–773.

Polson, P. G., Muncher, E., & Engelbeck, G. (1986). A test of a common elements theory of transfer. *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems* (pp. 78–83). New York: ACM.

Rieman, J., Young, R. M., & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies, 44*(6), 743–775.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Cognitive models as users. *ACM Transactions on Computer-Human Interaction, 7*, 141–173.

Ritter, F. E., & Young, R. M. (2001). Embodied models as simulated users: Introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human-Computer Studies, 55*, 1–14.

Salthouse, T. A. (1988). Initiating the formalization of theories of cognitive aging. *Psychology & Aging, 3*(1), 3–16.

Salvucci, D. D. (2001a). An integrated model of eye movements and visual encoding. *Cognitive Systems Research, 1*(4), 201–220.

Salvucci, D. D. (2001b). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies, 55*, 85–107.

Schoelles, M. J., & Gray, W. D. (2000). Argus Prime: Modeling emergent microstrategies in a complex simulated task environment. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modeling* (pp. 260–270). Veenendal, NL: Universal Press.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

Singley, M. K., & Anderson, J. R. (1989). *The transfer of cognitive skill.* Cambridge, MA: Harvard University Press.

St. Amant, R. (2000, Summer). Interface agents as surrogate users. *Intelligence* magazine, *11*(2), 29–38.

St. Amant, R., Horton, T. E., & Ritter, F. E. (in press). Model-based evaluation of expert cell phone menu interaction. *ACM Transactions on Human-Computer Interaction*.

St. Amant, R., & Riedl, M. O. (2001). A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies, 55*, 15–39.

St. Amant, R., Freed, A. R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research, 6*, 71–88.

Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model skill acquisition. *Human Factors, 45*, 61–76.

Tullis, T. S. (1983). The formatting of alphanumeric displays: A review and analysis. *Human Factors, 25*(6), 657–682.

Wiesmeyer, M. (1992). An operator-based model of covert visual attention. Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

Young, R. M., Barnard, P., Simon, T., & Whittington, J. (1989). How would your favorite user model cope with these scenarios? *ACM SIGCHI Bulletin, 20*, 51–55.

·6·

TASK LOADING AND STRESS IN HUMAN-COMPUTER INTERACTION: THEORETICAL FRAMEWORKS AND MITIGATION STRATEGIES

J. L. Szalma and P. A. Hancock
University of Central Florida

Tasks as Stressors: The Centrality of Stress in Human-Computer Interaction **116**
Traditional Approaches to Stress Research **116**
Theoretical Frameworks **116**
Appraisal Theory 116
Adaptation Under Stress 117
The Cognitive-Energetic Framework 118
Relationship between Stress and Cognitive Workload **120**
Cognitive Workload as a Form of Stress 120
Performance and Workload: Associations/Dissociations 120
Mitigation of Stress **120**
Changing the Person 120
Training/Skill Development 120
Personnel selection 120
Changing the Task 121
Display design 121
Adaptive automation 121
Hedonomics: Promoting Enjoyable Human-Computer Interaction 122
Teletic work and intrinsic motivation 122
Intrinsic motivation and skill development 122
Teletic work and motivational affordances 123
Problems for Future Research **123**
Understanding Mental Resources 123
Resource metaphors 123
Function of resources 124
Empirical tests of the model 124
Structural mechanisms 125
Energetic mechanisms 125
Structure and energy 126
Application of neuroergonomics 126
Development of the Adaptation under Stress Model . . . 126
Quantify the task dimensions 126
Attentional narrowing 126
Stressor characteristics 126
Potential benefits of space-time distortion 127
Changes in adaptation: the roles of time and intensity 127
Understanding Performance-Workload Associations/Dissociations 127
Task factors 127
Multidimensionality of workload 127
Individual Differences in Performance, Workload, and Stress **127**
Trait Differences 127
Affective State Differences 128
Attentional narrowing and adaptive response 129
Hedonomics and individual differences 129
Implications of Stress for Researchers and Practitioners **129**
Summary and Conclusions **129**
References **130**

TASKS AS STRESSORS: THE CENTRALITY OF STRESS IN HUMAN-COMPUTER INTERACTION

For those whose professional lives revolve around human-computer interaction (HCI), they may ask themselves why they should even glance at a chapter on stress. While it is evident that many computer systems have to support people operating in stressful circumstances, there are important design issues concerning how to present information in these very demanding circumstances. However, one could still ask, are these of central interest to those in the mainstream of HCI? Indeed, if these were the only issues, we would agree and would recommend the reader to pass quickly onto something of much more evident relevance. However, we hope to persuade you that the relevance of stress research to HCI is not limited to such concerns. Indeed, we hope to convince the reader that stress, in the form of task loading, is central to all HCI. To achieve this, we first present a perspective that puts stress front and center in the HCI realm. Traditionally, stress has been seen as exposure to some adverse environmental circumstances, such as excessive heat, cold, noise, vibration, and so forth, and its effects manifest themselves primarily in relation to the physiological system most perturbed by the stress at hand. However, Hancock and Warm (1989) observed that stress effects are virtually all mediated through the brain, but for the cortex such effects are almost always of secondary concern since the brain is primarily involved with the goals of ongoing behavior or, more simply, the current task. Therefore, we want to change the orientation of concern so that stress is not just a result of peripheral interference but rather that the primary source of stress comes from the ongoing task itself. If we now see the task itself as the primary driving influence then stress concerns are central to all HCI issues.

It is one of the most evident paradoxes of modern work that computer-based systems, which are designed to reduce task complexity and cognitive workload, actually often impose even greater demands and stresses on the very individuals they are supposed to be helping. How individuals cope with such stress has both immediate and protracted effects on their performance and well-being. Although operational environments and their associated tasks vary considerably (e.g., air traffic control, baggage screening, hospital patient monitoring, power plant operations, command and control, and banking and finance), there are certain mechanisms that are common to the stress appraisal of all task demands. Thus, there are design and HCI principles for stress that generalize across multiple domains (Hancock & Szalma, 2003a). In this chapter we explore such principles to understand stress effects in the HCI domain.

The structure of our chapter flows from these fundamental observations. First, we provide the reader with a brief overview of stress theory and its historical development to set our observations in context. Second, we articulate areas for future research needed to more completely understand how stress and workload impact human-computer interaction and how to exploit the positive effects while mitigating their negative effects.

TRADITIONAL APPROACHES TO STRESS RESEARCH

Traditionally, stress has been conceived of as either (a) an external, aversive stimulus (constituted of either physical, cognitive, or social stimulation patterns) imposed upon an individual or (b) response to such perturbations. Each of these views presents certain intrinsic operational difficulties. Considering stress as external stimulation is useful for categorizing effects of the physical environments (e.g., heat, noise, vibration), but such an approach cannot explain why the same stimulus pattern has vastly different effects on different individuals. Physiological interpretations (e.g., Selye, 1976) have tried to promulgate arousal explanations of stress. However, the more recent recognition that different sources of stress are associated with different patterns of cognitive effects made clear that adaptation or arousal theories of stress do not completely address the issue either (Hockey, R., 1984; Hockey, R. & Hamilton, 1983; Hockey, G. R. J., Gaillard, & Coles, 1986).

Thus to understand stress effects, we now have to embrace an even wider, multidimensional perspective (e.g., Matthews, 2001). Here we emphasize a view of stress as primarily an outcome of the appraisal of environmental demands as either taxing or exceeding an individual's resources to cope with that demand. These person-environment transactions (Lazarus & Folkman, 1984) occur at multiple levels within the organism (Matthews, 2001; van Reekum & Scherer, 1997). Further, these processes represent efforts by organism to adapt to demands imposed via regulation of both the internal state and the external environment. In the following section, we describe the theoretical frameworks that guide our observations on HCI. These perspectives emerge from the work of Hancock and Warm (1989), G. R. J. Hockey, (1997), and Lazarus (1999; see also Lazarus & Folkman, 1984).

THEORETICAL FRAMEWORKS

Appraisal Theory

Among the spectrum of cognitive theories of stress and emotion, perhaps the best known is the relational theory proposed by Richard Lazarus and his colleagues (see Lazarus, 1991, 1999; Lazarus & Folkman, 1984). This theory is cognitive in that stress and emotion each depend upon an individual's cognitive appraisals of internal and external events, and these appraisals depend in part on the person's knowledge and experience (cf., Bless, 2001). The theory is motivational in that emotions in general, including stress responses, are reactions to one's perceived state of progress toward or away from one's goals (see Carver & Scheier, 1998). The relational aspect emphasizes the importance of the transaction between individuals and their environment. Together these three components shape the emotional and stress state of an individual. The outcomes of these processes are patterns of appraisal that Lazarus (1991) referred to as "core relational themes." For instance, the core relational theme for anxiety is uncertainty and existential threat, while that

for happiness is evident progress toward goal achievement. Thus, when individuals appraise events relative to their desired outcomes (goals), these can produce negative, goal-incongruent emotions and stress if such events are appraised as hindering progress. Conversely, promotion of well-being and pleasure occur when events are appraised as facilitating progress toward a goal (e.g., goal-congruent emotions). Promotion of pleasure and happiness (see Hancock, Pepe, & Murphy, 2005; Ryan & Deci, 2001) therefore requires the design of environments and tasks themselves that afford goal-congruent emotions. The understanding of interface characteristics in HCI which facilitate positive appraisals and reduce negative appraisals is thus a crucial issue and an obvious avenue in which HCI and stress research can fruitfully interact.

A major limitation of all appraisal theories, however, is the neglect of understanding how task parameters influence resulting coping response. While the appraisal mechanism itself may be similar across individuals and contexts (e.g., see Scherer, 1999), the specific content (e.g., which events are appraised as a threat to well-being) does vary across individuals and contexts. One would expect that the criteria for appraisal (e.g., personal relevance, self-efficacy for coping) would be similar across individuals for specific task parameters as for any other stimulus or event. Individual differences occur in the specific content of the appraisal (e.g., one person's threat is another's challenge) and in the resultant response. An understanding of stress effects in HCI therefore requires understanding the task and person factors, and treating the transaction between the human and the system as the primary unit of analysis (see Lazarus & Folkman, 1984). This entails knowing how different individuals appraise specific task parameters and how changes in knowledge structures might ameliorate negative stress effects and promote positive affect in human-technology interaction. A visual representation of this emergent unit of analysis that comes from the interaction of person and environment, including the task, is shown in Fig. 6.1 (Hancock, 1997).

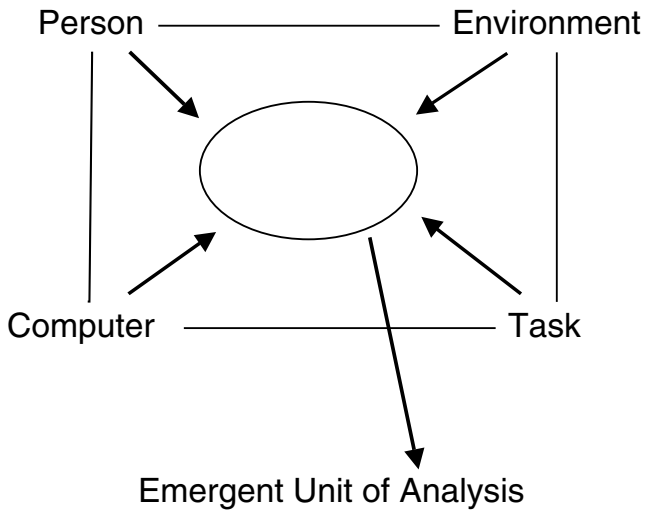


FIGURE 6.1. An illustration of the emergence of a supraordinate unit of analysis that derives from the interaction of the individual (person, the tool they use (computers), the task they have to perform, and the context (environment) in which action occurs. From Hancock 1997)

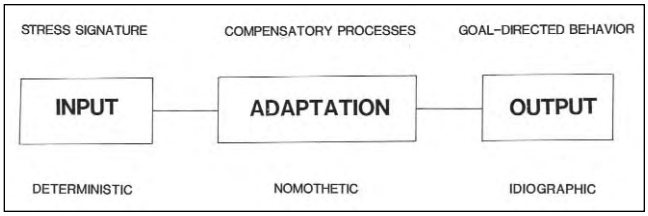


FIGURE 6.2. The trinity of stress which identifies three possible 'loci' of stress. It can be viewed as an *input* from the physical environment which can be described deterministically. Since such a profile is by definition unique, it is referred to as a stress signature. The second locus is *adaptation*, which describes the populational or nomothetic reaction to the input itself. It is most evidently measurable in the processes of compensation. The third and final locus is the *output* which is expressed as the impact on the on-going stream of behavior. Since the goals of different individuals almost always vary, this output is largely idiographic or person-specific. It is this facet of stress which has been very much neglected in prior and contemporary research. From Hancock and Warm (1989)

Adaptation Under Stress

A theoretical framework developed specifically for stress as it relates to performance is the maximal adaptability model presented by Hancock and Warm (1989). They distinguished the three facets of stress distinguished above, and they labeled these the trinity of stress, shown in Fig. 6.2. The first, "Input," refers to the environmental events to which the individual is exposed which include information (e.g., displays) as well as traditional input categories such as temperature, noise, vibration, and so forth (e.g., Conway, Szalma, & Hancock, 2007; Hancock, Ross, Szalma, in press; Pilcher, Nadler, & Busch, 2002). The second, "Adaptation," encompasses the appraisal mechanisms referred to previously. The third and final component, "Output," is the level which indicates how the organism behaves in respect to goal achievement. A fundamental tenet of the Hancock and Warm (1989) model is that in the large majority of situations (and even in situations of quite high demand) individuals do adapt effectively to the input disturbance. That is, they can tolerate high levels of either overload or underload without enormous change to their performance capacity. Adaptive processes occur at multiple levels, some being the physiological, behavioral (e.g., performance), and subjective/affective levels. These adaptations are represented in the model as a series of nested, extended, inverted-U functions (see Fig. 6.3) that reflect the fact that under most conditions the adaptive state of the organism is stable. However, under extremes of environmental underload or overload, failures in adaptation do occur. Thus, as the individual is perturbed by the input, the first threshold they traverse is subjective comfort. This is followed by behavioral effects, and finally failure of the physiological system (e.g., loss of consciousness). Examples of such extreme fail-

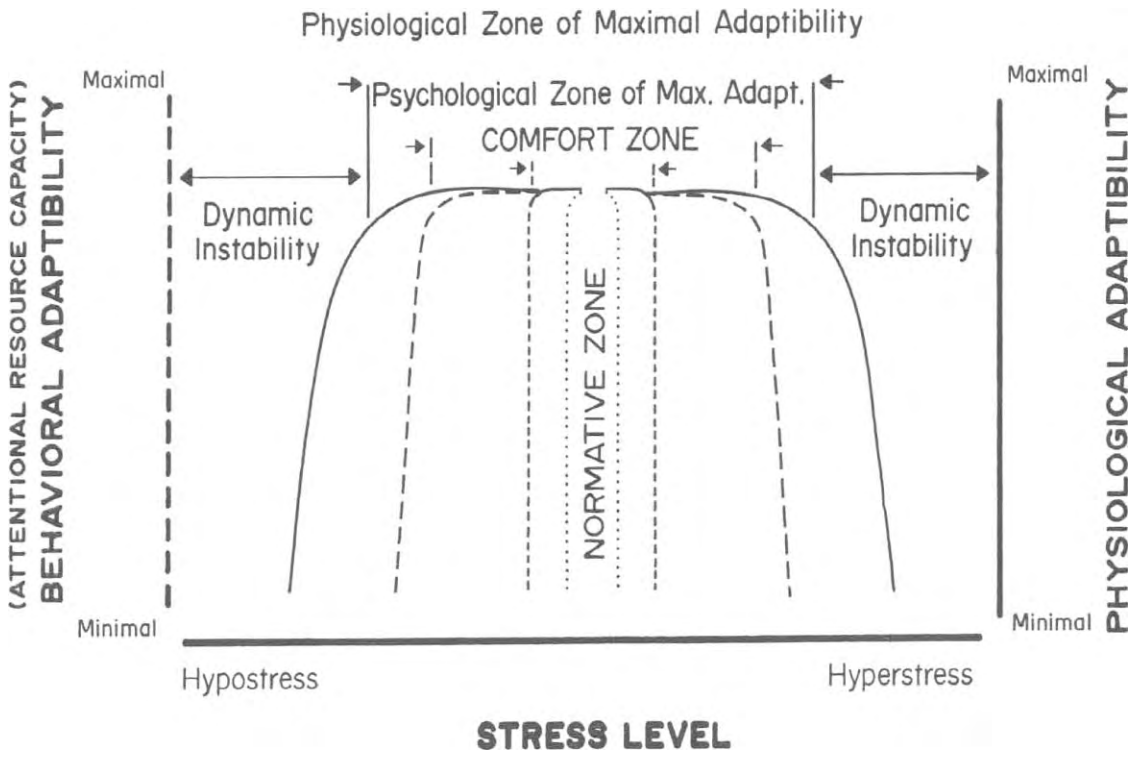


FIGURE 6.3. The extended-U relationship between stress level and response capacity. As is evident, the form of degradation is common across the different forms of response. At the center of the continuum is the normative zone which reflects optimal subjective functioning. Outside of this is the comfort zone which reflects the behavioral recognition of a state of satisfaction. Beyond this lies the reaction of psychological or cognitive performance capacity. Finally, the outer envelope is composed of physiological functioning. There are proposed strong linkages between the deviation from stability at one level being matched to the onset of radical failure at the more vulnerable level which is nested within it. The model is symmetrical in that underload (hypostress) has mirror effects to overload (hyperstress). The latter is considered the commonly perceived interpretation of stress. From Hancock and Warm (1989)

ures are relatively rare in most work settings, although when they do occur they are often catastrophic for the individual and the system they are operating (e.g., Harris, Hancock, & Harris, 2005).

This model is unique in that it provides explicit recognition that the proximal form of stress in almost all circumstances is the task itself. Task characteristics are incorporated in the model by two distinct base axes representing spatial and temporal components of any specified task. Information structure (the spatial dimension) represents how task elements are organized, including challenges to such psychological capacities such as working memory, attention, decision making, response capacity, and the like. The temporal dimension is represented by information rate. Together these dimensions can be used to form a vector (see Fig. 6.4) that serves to identify the current state of adaptation of the individual. Thus, if the combination of task characteristics and an individual's stress level can be specified, a vector representation can be used to predict behavioral and physiological adaptation. The challenge lies in quantifying the information processing components of cognitive work (see Hancock, Szalma, & Oron-Gilad, 2005).

Although the model shown in Fig. 6.4 describes the level of adaptive function, it does not articulate the mechanisms by which such adaptation occurs. Hancock and Warm (1989) argued that one way in which individuals adapt to stress is to narrow their attention by excluding task irrelevant cues (Easterbrook, 1959). Such effects are known to occur in spatial perception (e.g., Bursill, 1958; Cornsweet, 1969), and narrowing can occur at levels of both the central and peripheral neural systems (Dirkin & Hancock, 1984; 1985; Hancock & Dirkin, 1983). More recently Hancock and Weaver (2005) argued that distortions of temporal perception under stress are also related to this narrowing effect. However, recent evidence suggests that these two perceptual dimensions (space and time) may not share common perceptual mechanisms (see Ross, Szalma, Thropp, & Hancock, 2003; Thropp, Szalma, & Hancock, 2004).

The Cognitive-Energetic Framework

The Hancock and Warm (1989) model accounts for the levels of adaptation and adaptation changes under the driving forces

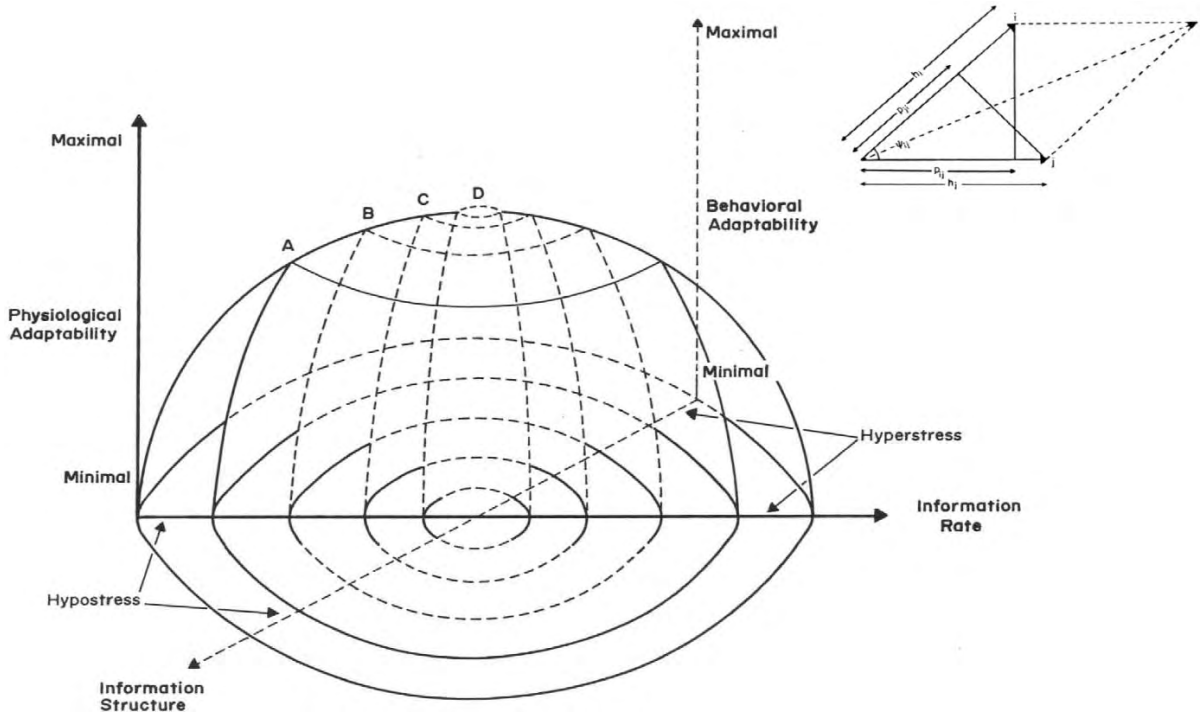


FIGURE 6.4. The description given in Figure 6.3 is now expanded into a three-dimensional representation by parsing the base "hypostress-hyperstress" axis into its two component elements. These divisions are compose of information rate (the temporal axis) and information structure (the spatial axis). Note that any one source of input stress can be described as a scalar on the base axis and these scalars can be summed to provide a multi-input stress vector which then provides a prediction of both performance and physiological adaptability which are the primary descriptors on the vertical axis. From Hancock and Warm (1989)

of stress. However, it does not articulate how effort is allocated under stress or the mechanisms by which individuals appraise the task parameters that are the proximal source of stress. The effort allocation issue is address by a cognitive-energetical framework described by G. R. J. Hockey (1997). The compensatory control model is based upon three assumptions: behavior is goal directed; self-regulatory processes control goal states; and regulatory activity has energetic costs (e.g., consumes resources). In this model a compensatory control mechanism allocates resources dynamically according to the goals of the individual and the environmental constraints. The mechanisms operate at two levels (see Fig. 6.5). The lower level is more or less automatic and represents established skills. Regulation at this level requires few energetic resources or active regulation and effort (cf., Schneider & Shiffrin, 1977). The upper level is a supervisory controller which can shift resources (effort) strategically to maintain adaptation and reflects effortful and controlled processing. The operation of the automatic lower loop is regulated by an effort monitor, which detects changes in the regulatory demands placed on the lower loop. When demand increases beyond the capacity of the lower loop control is shifted to the higher, controlled processing loop. Two strategic responses of the supervisory system are increased effort and changing the goals. Goals can be modified in their kind (change the goal itself) or in strength (e.g., lowering the criterion for performance). Essentially, this is adjusting the discrepancy between goal state and current state by increasing effort or changing the goal (and see Carver & Scheier, 1998).

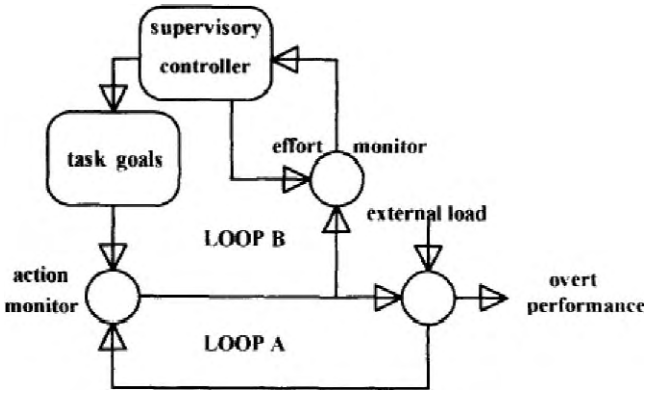


FIGURE 6.5. The two-level effort regulation model by Hockey. This model provides a mechanism by which an individual allocates limited cognitive resources to different aspects of performance. From Hockey (1997).

RELATIONSHIP BETWEEN STRESS AND COGNITIVE WORKLOAD

Cognitive Workload as a Form of Stress

The Hancock and Warm (1989) model explicitly identified the task itself as the proximal source of stress. In operational environments, this is often manifested as increases or decreases in cognitive workload (Moray, 1979). As in the case of stress, workload is easy to define anecdotally but difficult to define operationally. Workload can manifest in terms of the amount of information to be processed (an aspect of information structure), and the time available for processing (information rate). Thus, the base axes of the Hancock and Warm model captured dimensions of workload as well as stress (see Hancock & Caird, 1993). Indeed, physiological measures of workload (O'Donnell & Eggemeier, 1986) are often the same as those measures used to assess physiological stress. Similarly, subjective measures of workload and stress reflect appraisals of the task environment and of its perceived effect on the individual (Hart & Staveland, 1988). Although the two concepts developed in separate research traditions, the artificial boundary between them should be dissolved, as each term refers to similar processes. The implication for HCI is that computer-based tasks that impose either too much or too little demand will likely be appraised as stressful. Thus, the design process for development of computer interfaces should include assessment of perceived workload as well as affective state.

Performance and Workload: Associations/Dissociations

It is often the case that performance is maintained under increased workload and stress, which is reflected in the extended-U model described by Hancock and Warm (1989) and in the mechanisms of Hockey's (1997) energetic model of compensatory control. Maintaining performance under stress has costs, both physiologically and cognitively. Further, one would expect that in easier tasks performance is not as costly and that there should therefore be a direct association between task difficulty and perceived workload. Such performance-workload associations do occur, most prevalently in vigilance (Warm, Dember, & Hancock, 1996; see also Szalma et al., 2004). However, other forms of workload-performance relations can occur. For instance, perceived workload may change as a function of changes in task demand, but performance remains constant. Hancock (1996) referred to these situations as "insensitivities," which can be diagnostic with respect to the relation between the individual and the task (see also Parasuraman & Hancock, 2001). Thus, consistent with both Hancock and Warm and G. R. J. Hockey's (1997) frameworks, one response to increased task demand is to exert more effort, thereby maintaining performance but increasing perceived workload. Alternatively, one could have a situation in which task demands increase, performance decreases, but perceived workload does not change. This suggests that the appraisals of the task are not sensitive to actual changes in that task.

Interesting corollaries are the performance-workload dissociations that sometimes occur (Hancock, 1996; Yeh & Wickens, 1988). In such cases, decreased performance is accompanied by decreased workload. One possible reason for such a result might be disengagement of the individual from the task (e.g., the person gives up; see Hancock, 1996). In the case where increased performance is observed to be accompanied by increased perceived workload, the pattern suggests effective improvement of performance at the cost of increased effort allocation. An area of much needed research is establishing which task parameters control the patterns of performance-workload associations and dissociations, and how these change dynamically as a function of time on task. It may well be that reformulating the task by innovations in the interface itself may well address these crucial concerns (see Hancock, 1997). Indeed, the structure and organization of computer interfaces will be a major factor in both performance under stress and in the relation of performance to perceived workload.

MITIGATION OF STRESS

If changing the fundamental nature of the demand is one solution, we now look at other approaches to mitigation of the negative effects of stress and workload. These strategies include skill development (e.g., Hancock, 1986), and specific display design (Hancock & Szalma, 2003a; Wickens, 1996), as well as technologies employing adaptive automation and decision aids (Hancock & Chignell, 1987). Developing skills so that they are relatively automatic rather than controlled processing (Schneider & Shiffrin, 1977) and developing expertise can mitigate some of the negative effects of stress. In regard to display design, simple, easily perceivable graphics can permit quick, direct extraction of information when cognitive resources are reduced by stress and workload (Hancock & Szalma, 2003a). Adaptive automation can be employed by adjusting the level of automation and the management of automation according to stress state (e.g., Scerbo, Freeman, & Mikulka, 2003). In addition, adapting the form of automation (e.g., level, management type) to the operator based on individual differences can serve to improve its utility for aiding performance and reducing stress and workload (see Thropp, Oron-Gilad, Szalma, & Hancock, 2007).

Changing the Person

Training/skill development. Clearly, the greater the skill of the individual the more resilient their performance will be under stress (Hancock, 1986). This well-established phenomenon is incorporated into the energetic theories of stress and performance (Hancock & Warm, 1989; Hockey, 1997) and is an approach most often taken to mitigate workload and stress effects. However, training on relevant tasks is only one method of training for stress. There are also techniques for training individuals to cope more effectively with stress, essen-

tially building stress coping skills. An example of such an approach is stress exposure training (SET; Johnston & Cannon-Bowers, 1996), a three phase procedure in which individuals are provided with information regarding the stress associated with task performance, are provided with training on the task, and then practice their task skills under simulated stress conditions. This technique has been shown to be effective in reducing anxiety and enhancing performance (Saunders, Driskell, Johnston, & Salas, 1996) and there is preliminary evidence that coping skills learned with a particular type of stressor and task can transfer to novel stressors and tasks (Driskell, Johnston, & Salas, 2001). For such an intervention to succeed, however, it is crucial that the training be designed based on an analysis of the task environment (Johnston & Cannon-Bowers, 1996). If the task parameters that are most responsible for the workload and stress are identified, these can be targeted for attention in training.

An additional issue for training for more effective stress coping is to modify the individual's appraisal of events. By inducing automaticity in some skills, not only are more resources freed for stress coping, but the task environment itself will be appraised as less threatening. Even if the event is appraised as a threat to an individual's psychological or physical well-being, the highly skilled individual will appraise his or her coping ability as sufficient to handle the increased demand. However, there has been limited research on how individuals who develop expertise also develop the capacity to effectively cope with the stress that accompanies performance in a given domain, and the extent to which stress coping skills in one domain transfer to other domains. Deliberate practice generally facilitates skill development (Ericsson, 2007). If one considers coping with stress to be a skill, then in principle deliberate practice should permit the development of expertise in coping. This will likely involve parsing the task into components, based on cognitive task analysis, and designing training procedures that target the stressful aspects of the task. However, such efforts will require understanding of how different forms of stress affect different forms of information processing. Since these variables are difficult to quantify, establishing these linkages must be theory driven. Elucidation of these issues will provide the groundwork for future development of stress mitigation tools during training and skill development.

Personnel selection. Selection techniques have been a popular choice for matching individuals to specific jobs, but the focus has typically been on intellectual skills (e.g., Yerkes, 1918). Selecting individuals for their stress-coping capability has been applied to the selection criteria for police officers, who therefore tend to be as stable as or more emotionally stable than the rest of the population (for a review, see Brown & Campbell, 1994). However, research is needed that links particular traits to stress coping skills for specific task environments. The effectiveness of general life stress coping, such as that observed in individuals who are extraverted (McCrae & Costa, 1986; Penley & Tomaka, 2002) or optimistic (Aspinwall, Richter, & Hoffman, 2002; Scheier & Carver, 1985), may not predict effective coping in specific task domains. Understanding which individuals will likely cope effectively with a particular task therefore requires a thorough understanding of the perceptual, cognitive, and psychomotor demands of the task, and then linking these parameters to trait profiles. By far, the most research on the relation of affective traits to task performance has been in Extraversion and trait anxiety/Neuroticism (for a review, see Matthews, Deary, & Whiteman, 2003). However, the characteristics of greatest interest may vary somewhat across domains, although some general traits (e.g., emotional stability, conscientiousness) would be expected to moderate performance across a variety of task environments.

Changing the Task

Display design. Although training and selection can mitigate stress effects, the tasks themselves should be redesigned, for two reasons. First, there will be many instances where selection is not possible and expenditure of significant resources on training is undesirable. Second, there are instances in which one wishes to design an interface that requires little or no training and that can be used by any member of a large population of individuals (e.g., consumers). Particularly in light of the observation that the task represents the proximal source of stress, future work in stress mitigation for HCI should focus on redesign of the task and the interface itself. We have previously argued that existing display design techniques that are simple and easily perceived would be the best choice for an interface that will be used in stressful environments (Hancock & Szalma, 2003a). Specifically, configural or object displays can represent complex, multivariable systems as simple geometric shapes or emergent features if those features are well-mapped to system dynamics (see Bennett & Flach, 1992). Under stress, the complex problem solving and analytical skills are the most vulnerable and decline first. A display that allows fast extraction of information with minimal cost in working memory load can mitigate stress effects (Hancock & Szalma, 2003a; Wickens, 1996). A combination of training to automaticity and displays of information that can be perceived directly with a minimum of information processing requirements is currently one of the best approaches for stress mitigation in cognitively complex environments.

Adaptive automation. Another approach for stress mitigation is the allocation of function to automated systems (Hancock & Chignell, 1987). The advent of modern automated systems allows for automation to adapt to the state of the individual (Scerbo et al., 2003). Thus, at points in time when an operator is overtaxed, the system can assume control of some task functions, thereby freeing resources to effectively cope with increased task demand. Two potential problems for automated systems are that over reliance can occur and operator skills can atrophy. However, a dynamic (adaptive) automated system that permitted or required the operator to perform functions at different points in time could reduce the probability of skill atrophy while still relieving the workload and stress of task performance.

However, the introduction of automation can itself induce stress. Operators who work with automated systems, particularly static, inflexible automation, are relegated to the role of monitors who must respond only when untoward events occur. Sustained

attention requirements are in fact quite stressful (Szalma, 1999; Warm, 1993), and paradoxically induce higher levels of perceived workload (Warm et al., 1996). Adaptive automation can mitigate this problem by dynamically assigning tasks to the machine or the human depending on the environmental conditions and the state of the operator (Hancock & Chignell, 1987). Indeed, efforts to use operator neurological state to adjust automation are currently underway (e.g., Scerbo, 2007).

Hedonomics: Promoting Enjoyable Human-Computer Interaction

Stress research has traditionally followed the edict of ergonomics and human factors in general, to do no harm and to prevent pain and injury. As with the rest of behavioral science, stress researchers sought to treat the symptoms of stress and mitigate its negative effects on performance. However, with the advent of positive psychology (Seligman & Csikszentmihalyi, 2000) there has been a movement to incorporate the promotion of pleasure and well-being rather than restricting efforts to pain prevention. Hancock coined the term *hedonomics* and defined it as that branch of science that facilitates the pleasant or enjoyable aspects of human-technology interaction (Hancock, Pepe, & Murphy, 2005). In short, the goal for hedonomics is to design for happiness. Hedonomics is a fairly new research area, but during the last 10 years, there has been a rapid growth in research concerning affect and pleasure. Affective evaluations provide a new and different perspective in Human Factors Engineering. It is not how to evaluate users—it is how the user evaluates (Hancock, Pepe, et al., 2005). The research on hedonic values and seductive interfaces is in fact a welcome contrast to safety and productivity, which have dominated human factors and ergonomics. Note, however, that pleasurable interaction with technology is not necessarily conducive to happiness. Indulging pleasures can sometimes interfere with happiness and well-being (see Fromm, 1976; Kasser, 2002; Ryan & Deci, 2001).

Our argument is not that we should discard current methods in human factors and ergonomics. Clearly functionality and usability are necessary conditions for pleasurable interaction with technology. If an interface does not function in a way congruent with the user's goals, so that the user appraises the technology as an agent that is interfering with goal achievement, that interaction is likely to be stressful and system performance more vulnerable to decline. However, function and usability are not sufficient conditions for pleasurable interactions with technology. The interface should be designed such that it affords appraisals of the technology as a convivial tool (Illich, 1973) or aid. One can also utilize the human tendency to anthropomorphize technology to facilitate appraisals of the technology as helpful and supportive rather than as an enemy (Luczak, Roetting, & Schmidt, 2003).

Hedonomic design will be of obvious importance for development of consumer products, but in principle, it can also transform the very nature of work, rendering it fun. Although there will be some tasks which will never be enjoyable, there are many individuals who have jobs that could be made more enjoyable by designing the tasks such that they promote teletic work (Csikszentmihalyi, 1990) and facilitate intrinsic motivation (Deci & Ryan, 2000).

Teletic work and intrinsic motivation. A useful theoretical framework for hedonomics is Self-Determination Theory (SDT; Deci & Ryan, 1985, 2000; Ryan & Deci, 2000, 2001). From this perspective there are three organismic needs that are essential for facilitating intrinsic motivation for task activity and the positive affect that can accompany such states. These needs are for competence (self-efficacy; see also Bandura, 1997), autonomy (personal agency, not independence per se), and relatedness. An important difference between this theory and other theories of motivation is the recognition that there are qualitatively different forms of motivation (Gagne & Deci, 2005). Thus, in SDT five categories of motivated behavior are identified that vary in the degree to which the motivation is self-determined. Four of the categories reflect extrinsic motivation and one category is intrinsic motivation. In the latter case, an individual is inherently motivated to engage in activity for its own sake or for the novelty and challenge. The four extrinsic motivation categories vary in the degree to which regulation of behavior is internalized by the individual and therefore more autonomous and self-determined (Ryan & Deci, 2000). The process of internalization involves transforming an external regulation or value into one that matches one's own values. The development of such autonomous motivation is crucial to skill development, since the person must maintain effort throughout a long and arduous process. Individuals who are autonomously motivated to learn are those who develop a variety of effective self-regulation strategies, have high self-efficacy, and who set a number of goals for themselves (Zimmerman, 2000). Further, effective self-regulation develops in four stages: observation, emulation, self-control, and self-regulation. Successful skill development involves focus on process goals in early stages of learning and outcome goals in the fourth stage (Zimmerman, 2000).

Intrinsic motivation and skill development. Research has established that intrinsic motivation is facilitated by conditions promoting autonomy, competence, and relatedness (see Deci & Ryan, 2000). Three factors that support autonomy are (a) meaningful rationale for doing the task, (b) acknowledgement that the task might not be interesting, and (c) an emphasis on choice rather than control. It is important to note that externally regulated motivation predicts poorer performance on heuristic tasks (Gagne & Deci, 2005), suggesting that as experts develop better knowledge representations it will be crucial to promote internal regulation of motivation. Although intrinsic motivation has been linked to how task activities and environmental contexts meet psychological needs, it is not clear why skilled performers are able to meet these needs, or why individuals choose a particular computer interface. It is likely that interest in activities codevelops with abilities and traits (see Ackerman & Heggestad, 1997), but this issue needs thorough investigation in the context of complex computer environments that require highly skilled workers.

In addition to the issue of efficacy and self-regulation, there is a need to examine the process by which individuals internalize extrinsic motivation as they gain experience with a particular interface or system. In particular, Gagne and Deci (2005) noted that little research has examined the effect of reward structures and work environments on the internalization process. It is likely that those environments that are structured to meet ba-

sic needs will more likely facilitate internalization processes and inoculate learners against the trials and tribulations that face them as they interact with new technologies.

Teletic work and motivational affordances. Teletic, or autoteletic, work refers to "work" that is experienced as enjoyable and is associated with flow or optimal experience characterized by a sense of well being and harmony with one's surroundings (Csikszentmihalyi, 1990). There is variation in both tasks and individuals with respect to the degree to which the human-technology interaction is teletic. There are four categories in which individuals tend to fall with respect to their relation to work. First, there is a small proportion of the population that are always happy in life, regardless of their activity, which Csikszentmihalyi referred to as individuals with an autotelic personality. There are also individuals who are predisposed to happiness about a specific task. They appraise such tasks as enjoyable, and seek out these activities. The third group consists of individuals who enjoy specific activities but cannot do them professionally, such as amateur athletes. The vast majority of people, however, do work for purely functional reasons (e.g., security). For these individuals work is boring and grinding because the task itself is aversive. A goal for hedonomics, then, is to design work that can be enjoyed to the greatest extent possible. This means structuring the environment as an entire system, ranging from the specific cognitive and psychomotor demands to the organization in which the person works. Even in jobs that are not inherently enjoyable, some degree of positive affect can be experienced by workers if they their environment is structured to facilitate a sense of autonomy (personal agency), competence, and relatedness (Deci & Ryan, 2000; see also Gagne & Deci, 2005). From an ecological perspective (e.g., Flach, Hancock, Caird, & Vicente, 1995), this means identifying the *motivational affordances* in the task and work environment, and designing for these affordances. Thus, just as one might analyze the affordance structure of an interface using ecological interface design methods (e.g., Vicente & Rasmussen, 1992), one can design an environment so that the elements of the physical and social environment afford stress reduction and enhanced intrinsic motivation. Note that although an affordance is a property of the environment, it does not exist independently of the individual in the environment. Affordances therefore share conceptual elements of person-environment transactions that drive emotion and stress. They differ in that the classical definition of an affordance is a physical property of the environment (Gibson, 1966, 1979), while a transaction emphasizes the individual's subjective appraisal of the environment. In both cases, however, one cannot define the concept by isolating either the individual or the context. Thus, although affordances and invariants are considered physical properties of the environment, these concepts are still relevant for motivational processes (and see Reed, 1996).

Motivational affordances may be conceived as elements of the work environment that facilitate and nurture intrinsic motivation. The key for design is to identify motivational invariants or environmental factors that consistently determine an individual's level of intrinsic motivation across contexts. There are some aspects of work that have been identified as important for facilitating intrinsic motivation and would thus be considered motivational invariants. For instance, providing feedback that is perceived as controlling rather than informative tends to undermine a sense of autonomy and competence and thereby reduces intrinsic motivation (Deci, Ryan, & Koestner, 1999). Careful analyses of the motivational affordance structure will permit design of tasks that are more likely to be enjoyable by rendering the tools convivial (Illich, 1973) and thereby facilitating human-machine synergy (and see Hancock, 1997).

PROBLEMS FOR FUTURE RESEARCH

In this section, we will identify the areas for future research. These include a better understanding of resources and quantifying task dimensions defined in the Hancock and Warm (1989) model, which will likely reduce to the thorny problem of quantifying human information processing (see Hancock, Szalma, & Oron-Gilad, 2005). Further, we will discuss the need for research on performance-workload associations and dissociations, and the evident need for programmatic investigation of individual differences in performance, workload, and stress.

The Hancock and Warm (1989) model of stress explicitly identified task dimensions that influence stress state and behavioral adaptability. However, the metrics for these dimensions, and how specific task characteristics map to them, have yet to be fully understood. Thus, future research should aim to examine how different task components relate to performance and subjective and physiological state. Development of a quantitative model of task characteristics will permit the derivation of vectors for the prediction of adaptability under stress. Cognitive Neuroscience and Neuroergonomics in particular offer one promising approach to such development. An additional step in this direction, however, will be facilitated by improved quantitative models of how humans process information (Hancock, Szalma, & Oron-Gilad, 2005).

Understanding Mental Resources

One of the challenges for quantifying human information processing is that there is little understanding or consensus regarding the capacities that process the information. A central concept in energetic models of human performance is mental resources. Resource theory replaced arousal and served as an intervening variable to explain the relations between task demand and performance. However, a continual problem for the resource concept is to operationally define what it is. Most treatments of resources use that term metaphorically (Navon & Gopher, 1979; Wickens, 1980, 1984), and failures to specify what resources are have led some to challenge the utility of the concept (Navon, 1984). As resource theory is a central concept in the theories of stress discussed herein, and represents one of the most important issues to be resolved in future research on stress and performance, we now turn to the definitional concerns associated with the resource construct and the imperative for future research to refine the concept.

Resource metaphors. Two general categories of resource metaphors may be identified: structural metaphors and ener-

getic metaphors. One of the earliest conceptualizations of resource capacity used a computer-based metaphor (Moray, 1967). Thus, cognitive capacity was viewed as analogous to the RAM and processing chip of a computer, consisting of information processing units that can be deployed for task performance. However, the structural metaphor has been applied more to theories of working memory than to attention and resource theory.[1] Most early resource theories, including Kahneman's (1973) original view and modifications by Norman and Bobrow (1975), Navon and Gopher (1979), and Wickens (1980, 1984), applied energetic metaphors to resources, and conceptualized them as commodities or as pools of energy to be spent on task performance. In general, energetic approaches tend to employ either economic or thermodynamic/hydraulic metaphors. The economic model is reflected in the description of resources in terms of supply and demand: Performance on one or more tasks suffers when the resource demands of the tasks exceed available supply. Presumably, the total amount of this supply fluctuates with the state of the individual, with the assets diminishing with increases in the intensity or duration of stress. Although Kahneman's original conception allowed for dynamic variation available resource capacity, most early models assumed a fixed amount of resources (see Navon & Gopher, 1979). In thermodynamic analogies, resources are a fuel that is consumed, or a tank of liquid to be divided among several tasks, and under stressful conditions the amount of resources available is depleted and performance suffers. In discussing his version of resource theory, Wickens (1984) warned that the hydraulic metaphor should not be taken too literally, but most subsequent descriptions of resources have employed visual representations of resources as just this form (e.g., a tank of liquid). Similarly, many discussions of resource availability and expenditure adopt the economic language of supply and demand, and Navon and Gopher explicitly adopted principles of microeconomics in developing their approach. An additional problem for resource theory is that in most cases (e.g., Navon & Gopher, 1979; Wickens, 1980, 1984), the structural and energetic metaphors were treated as interchangeable, a further testament to the ambiguity of the construct.

A problem with using nonbiological metaphors to represent biological systems is that such models often fail to capture the complexity and the unique dynamic characteristics (e.g. adaptive responses) of living systems. For instance, a hydraulic model of resources links the activity of a tank of liquid, governed by thermodynamic principles, to the action of arousal mechanisms or energy reserves that are allocated for task performance. However, a thermodynamic description of the physiological processes underlying resources is at a level of explanation that may not adequately describe the psychological processes that govern performance. Thermodynamic principles can be applied to the chemical processes that occur within and between neurons, but they may be less useful in describing the behavior of large networks of neurons.[2] Similarly, economic metaphors of supply and demand may not adequately capture the relation between cognitive architecture and energy allocated for their function. Economic models of resources define them as commodities to be spent on one or more activities, and they assume an isomorphism between human cognitive activity and economic activity, an assumption which may not be tenable. Indeed, Navon and Gopher (1979) admitted that their static economic metaphor for multiple resources may need to be replaced by a dynamic one that includes temporal factors (e.g. serial versus parallel processing; activity of one processing unit being contingent upon the output of another). Such concerns over the metaphors used to describe resources are hardly new (Navon, 1984; Wickens, 1984), but their use has become sufficiently ingrained in thinking about resources and human performance that reevaluation of the metaphors is warranted. A regulatory model based on physiology may serve as a better metaphor (and, in the future may serve to describe resources themselves to the extent that they can be established as a hypothetical construct) to describe the role of resources in human cognition and performance. However, even a physiologically-based theory of resources must be tempered by the problems inherent in reducing psychological processes to physiological activity.

Function of resources. Another problem for resource theory is the absence of a precise description of how resources control different forms of information processing. Do resources determine the energy allocated to an information processor (Kahneman, 1973), do they provide the space within which the processing structure works (Moray, 1967), or does the processor draw on the resources as needed (and available)? In the latter case, the cognitive architecture would drive energy consumption and allocation, but the locus of control for the division of resources remains unspecified in any case. Presumably, an executive function that either coordinates information processors drawing on different pools of resources or decides how resources will be allocated must itself consume resources, in terms of both energy required for decision making and mental space or structure required. Hence, resource theory does not solve the homunculus problem for theories of attention, nor does it adequately describe resource allocation strategies behind performance of information processing tasks.

Empirical tests of the model. Navon and Gopher (1979) commented on the problem of empirically distinguishing declines in performance due to insufficient supply from those resulting from increases in demand. They asked, "When the performance of a task deteriorates, is it because the task now gets fewer resources or because it now requires more?" (p. 243). Navon and Gopher characterized the problem as distinguishing between changes in resources and changes in the subject-task parameters that constrain resource utilization, and they offered two approaches to avoid this difficulty. One approach is to define the fixed constraints of the task and observe how the information processing system manages the processes within those constraints. The degree of freedom of the system, in this view,

[1]This is a curious historical development, since these relatively separate areas of research converge on the same psychological processes.

[2]The argument here is not that neural structures are not constrained by the laws of thermodynamics—clearly they are—but that thermodynamic principles implied by the metaphor are not sufficient for the development of a complete description of resources and their relation to cognitive activity.

is the pool of resources available, in which the term *resource* is interpreted broadly to include quality of information, number of extracted features, or visual resolution. The subject-task parameters define what is imposed on the system (the demands) and the resources refer to what the system does in response to the demands (allocation of processing units). From this perspective resources can be manipulated by the information processing system within the constraints set by the subject-task parameters. A second approach is to distinguish the kind of control the system exerts on resources, between control on the use of processing devices (what we have called "structure") and the control of the properties of the inputs that go into these devices. The devices are processing resources. The other kind of control is exerted on input resources, which represents the flexibility the person has for determining which inputs are operated on, as determined by subject-task parameters. Processing resources are limited by the capacities of the information processors, while the input resources are limited by subject-task parameters (and allocation strategies that determine which information the operator attends to). Presumably, the individual would have some control over the allocation strategy, in terms of the processing resources devoted to a task, although these can also be driven by task demands (e.g. a spatial task requires spatial processing units). Navon and Gopher did not advocate either approach, but presented them as alternatives for further investigation. The implication for examining the resource model of stress is that one must manipulate both the subject-task parameters (e.g. by varying the psychophysical properties of the stimulus, manipulating the state of the observer, or varying the kind of information processing demanded by the task) as well as the allocation strategies the operator uses (the input resources—e.g. payoff matrices, task instructions). This would provide information regarding how specific stressors impair specific information processing units and how they change the user's resource allocation strategies in the presence of stress that is continuously imposed on operators of complex computer-based systems.

In a later article, Navon (1984) moved to a position less favorable toward resources than the earlier approach, asserting that predictions derived by resource theory could be made, and results explained, without appealing to the resource concept (see also Rugg, 1986). One could instead interpret effects in terms of the outputs of information processors. Most manipulations, such as difficulty (which in his view influences the efficiency of a unit of resources) or complexity (which affects the load, or the number of operations required) influence the demand for processing, with supply having no impact upon their interaction. However, this approach assumes a clear distinction between outputs of a processing system and the concept of a resource, and Navon's notion of specific processors seems blurred with the notion of a resource, as both are utilized for task performance. Nevertheless, his critique that the regarding the vagueness of the resource concept is relevant, and Navon argued that if resources are viewed as an intervening variable rather than a hypothetical construct, the concept has utility for describing the process.

Structural mechanisms. If different kinds of information processing draw on different kinds of resources, in terms of the information processors engaged in a task, stressors may have characteristic effects on each resource. In addition, as Navon and Gopher (1979) noted, an aspect of resource utilization is the efficiency of each resource unit. It may be that stress degrades the efficiency of information processing units, independent of energy level or allocation strategy (cf., Eysenck, M. W. & Calvo, 1992). Investigation of such effects could be accomplished by transitioning between tasks requiring different kinds of information processing and determining if the effects of stress on one structure impacts the efficiency of a second structure.

The quality of resources can vary not only in terms of the kind of information processing unit engaged, but also in terms of the kind of task required. Following Rasmussen's (1983) classification system for behavior as a heuristic for design, some tasks require knowledge-based processing, in which the operator must consciously rely on his or her mental model of the system in order to achieve successful performance. Other tasks fall under the category of rule-based behavior, in which a set rules or procedures define task performance. The third category is skill-based behavior, in which the task is performed with a high degree of automaticity. Presumably, each kind of task requires different amounts of resources, but they may also represent qualitatively different forms of resource utilization. In other words, these tasks may differ in the efficiency of a unit of resources as well as different effort allocation strategies. As task performance moves from knowledge to rule to skill based processing (e.g. with training), the cognitive architecture may change such that fewer information processing units are required, and those that are engaged become more efficient. Moreover, the way in which each of these systems degrade with time under stress may be systematic, with the more fragile knowledge-based processing degrading first, followed by rule based processing, with skill based processing degrading last (at this point, one may begin to see breakdown of not only psychological processes but physiological ones as well; see Hancock & Warm, 1989). This degradation may follow a hysteresis function, such that a precipitous decline in performance occurs as the operator's resource capacity is reduced below a minimum threshold for performance. Moreover, these processes may recover in an inverse form, with skill-based processing recovering first, followed by rule and knowledge-based processing.

Note that it may be difficult to distinguish pure knowledge-based processing from rule- or skill-based activity. An alternative formulation is the distinction between controlled and automatic processing (Schneider & Shiffrin, 1977). Although originally conceived as categories, it is likely that individuals engaged in real-world tasks utilize both automatic and controlled processing for different aspects of performance and that for a given task there are levels of automaticity possible. Treating skills as a continuum rather than as discrete categories may be a more theoretically useful framework for quantifying resources and information processing, and thereby elucidating the effects of stress on performance.

Energetic mechanisms. To investigate the energetic aspects of resources, one must manipulate environmentally-based perturbations, in the form of external stressors (noise, heat) and task demands, to systematically affect inflow versus

outflow of energy. Presumably, inflow is controlled by arousal levels, physiological energy reserves, and effort. One could examine performance under manipulations of energetic resources under dual task performance (e.g., What happens to performance on two tasks under sleep deprivation or caffeine consumption?). For example, the steady state can be perturbed by increasing (e.g. caffeine) or decreasing (e.g. sleep deprivation) energy while systematically varying the demands for two tasks.

Structure and energy. Another empirical challenge is to distinguish resources as structure from resources as energy. Given the definitional problems associated with the resource concept, it is not clear whether performance declines because of reduction in energy level or degradation in structures (e.g., failures or declines in the efficiency of the processing units), or a combination of both. If structure and energy are distinct elements of resources, it is hypothetically possible to manipulate one while holding the other constant, although the validity of that assumption is questionable. Is it possible to manipulate specific forms of information processing under constant energy level? Is it possible to manipulate energy level independent of which cognitive processes are utilized? If the decline in available resources is, at least in part, due to the degradation of particular information processing units, then transferring to a task requiring the same processor should lead to worse performance than transferring to one that is different (cf., Wickens, 1980, 1984). For instance, if a person engages in a task requiring verbal working memory while under stress, then transitions to a task requiring spatial discrimination, performance on the latter should depend only on energetic factors, not on structural ones. Note, however, that in this case the effects of different mental capacities would be confounded with the effects of novelty and motivation on performance.

Application of neuroergonomics. The burgeoning field of Neuroergonomics seeks to identify the neural bases of psychological processes involved in real-world human-technology interaction (Parasuraman, 2003). As we have stated elsewhere (Hancock & Szalma, 2007), recent advances in Neuroergonimcs promises to identify cognitive processes and their link to neurological processes. This may permit a more robust and quantitative definition of resources, although we caution that a reductionistic approach is not likely to be fruitful (and see Hancock & Szalma, 2003b). In addition, the stress concept itself rests in part on more precise definitions of resources (Hancock & Szalma, 2007). Thus, resolution of the resource issue in regard to cognitive processing and task performance would also clarify the workload and stress concepts. We view Neuroergonomics as one promising avenue for future research to refine the workload and stress and resource concepts.

Development of the Adaptation under Stress Model

Quantify the task dimensions. A major challenge for the Hancock and Warm (1989) model is the quantification of the base axes representing task dimensions. Specification of these dimensions is necessary if the vector representation postulated by Hancock and Warm is to be developed and if the resource construct is to be more precisely defined and quantified. However, task taxonomies that are general across domains present a theoretical challenge, because they require an understanding and quantification of how individuals process information along the spatial and temporal task dimensions, and how these change under stressful conditions. Quantification of information processing, and subsequent quantification of the base axes in the Hancock and Warm model, permit the formalization of the vector representation of adaptive state under stress (see Fig. 6.4).

Attentional narrowing. Recall that Hancock and Weaver (2005) argued that the distortions of spatial and temporal perception have a common attentional mechanism. Two implications of this assertion are (a) that events (internal or external) that distort one dimension will distort the other, and (b) that these distortions are unlikely to be orthogonal. With very few exceptions, little research has addressed the possibility of an interaction between distortions of spatial and temporal perceptions in stressful situations on operator performance. Preliminary evidence suggests that these two dimensions may in fact not share a common mechanism (Ross et al., 2003; Thropp et al., 2004), although further research is needed to confirm these findings. An additional important issue for empirical research is whether we are dealing with time-in-memory or time-in-passing (and to some extent space-in-memory vs. space-in-passing). Thus, the way in which perceptions of space and time interact to influence operator state will depend upon how temporal perceptions (and spatial perception, for that matter) are measured.

A possible explanation for perceptual distortions under conditions of heavy workload and stress concerns the failure to switch tasks when appropriate. Switching failures may be responsible for the observation in secondary task methodology that some participants have difficulty dividing their time between tasks as instructed (e.g., 70% to the primary task and 30% to the secondary task). This difficulty may result from the participant's inability to accurately judge how long he or she has attended to each task during a given time period. The degree to which distortions in perception of space-time are related to impairments in task switching under stressful conditions, and the degree to which these distortions are related to attention allocation strategies in a secondary task paradigm, are questions for empirical resolution.

Stressor characteristics. Even if space and time do possess a common mechanism, it may be that specific stressors do not affect spatial and temporal perceptions in the same way. For instance, heat and noise may distort perception of both space and time, but not to the same degree or in the same fashion. It is important to note that spatial and temporal distortions may be appraised as stressful, as they might interfere with the information processing requirements of a task. Consequently, some kinds of information processing might be more vulnerable to one or the other kind of perceptual distortion. Clearly, performance on tasks requiring spatial abilities, such as mental rotation, could suffer as a result of spatial distortion, but they might be unaffected (or, in some cases, facilitated) by temporal distortion. Other tasks, such as those that rely heavily on working memory, mathematical ability, or tasks requiring target detec-

tion, could each show different patterns of change in response to space-time distortion.

Potential benefits of space-time distortion. Under certain conditions, the narrowing of spatial attention can benefit performance through the elimination of irrelevant cues. The precise conditions under which this occurs, however, remains unclear. In addition, it is important to identify the circumstances under which time distortion might actually prove beneficial. Here, operators perceive that they have additional time to complete the task at hand (Hancock & Weaver, 2005). This would have great benefit in task performance situations where attentional narrowing is less likely to have deleterious effects. At this point, this is an empirical question that might be amenable to controlled testing.

Changes in adaptation: the roles of time and intensity. The degree to which a task or the physical and social environment imposes stress is moderated by the characteristics of the stimuli as well as the context in which events occur. However, two factors that seem to ubiquitously influence how much stress impairs adaptation are the (appraised) intensity of the stressor and the duration of exposure. We have recently reported meta-analytic evidence that these two factors jointly impact task performance across different orders of task (e.g., vigilance, problem solving, tracking; see Hancock, Ross, & Szalma, in press). Duration is further implicated in information processing itself, and may be a central organizing principle for information processing in the brain (Hancock, Szalma, & Oron-Gilad, 2005). Empirical research is needed, however, to programmatically explore the interactive effects of these two variables across multiple forms of information processing.

Understanding Performance-Workload Associations/Dissociations

Task factors. Although Hancock (1996) and Yeh and Wickens (1988) articulated the patterns of performance-workload relations and how these are diagnostic with respect to processing requirements, there has been little systematic effort to further investigate these associations/dissociations. The primary question is what factors drive dissociations and insensitivities when they occur. For instance, for vigilance mostly associations are observed, while for other tasks, such as those with high working memory demand, dissociations are more common (Yeh & Wickens, 1988). Enhanced understanding of these relations would inform the Hancock and Warm (1989) model by permitting specification of the conditions under which individuals pass over the thresholds of failure at each level of person-environment transaction/adaptation.

Multidimensionality of workload. To date, consideration of performance-workload dissociations has been primarily concerned with global measures of perceived workload. However, there is clear evidence that perceived workload is in fact multidimensional. For instance, vigilance tasks are characterized by high levels of mental demand and frustration (Warm, Dember, & Hancock, 1996). It is likely that the pattern of performance-workload links will be different for different orders of performance (different tasks) but also for different dimensions of workload. One approach to addressing this question would be to systematically manipulate combinations of these two variables. For instance, if we consider performance in terms of detection sensitivity, memory accuracy, speed of response, and consider the dimensions of workload defined by the NASA Task Load Index (Hart & Staveland, 1988), one could examine how variations in memory load or discrimination difficulty link to each subscale.

INDIVIDUAL DIFFERENCES IN PERFORMANCE, WORKLOAD, AND STRESS

Elsewhere, we reviewed the relations between individual differences in state and trait to efforts to quantify human information processing (Szalma & Hancock, 2005). Here, we address how individual differences (state and trait) are related to stress and coping.

Trait Differences

Individual differences research has been a relatively neglected area in human factors and experimental psychology. Much of the early work on individual differences was done by individuals not concerned with human-technology interaction, to the extent that a bifurcation between two kinds of psychology occurred (Cronbach, 1957). There is evidence, however, that affective traits influence information processing and performance. Thus, extraversion is associated with superior performance in working memory tasks and divided attention, but also with poorer sustained attention (cf., Koelega, 1992). Trait anxiety is associated with poorer performance, although results vary across task types and contexts (Matthews et al., 2003). A possible next step for such research will be to systematically vary task elements, as discussed previously in the context of the Hancock and Warm (1989) model, and test hypotheses regarding how trait anxiety relates to specific task components (for an example applied to Extraversion, see Matthews, 1992). The theoretical challenge for such an undertaking is that it requires a good taxonomic scheme for tasks as well as a well-articulated theory of traits and performance. However, trait theories have neglected specific task performance, focusing instead on global measures (e.g., see Barrick, Mount, & Judge, 2001), and there is a lack of a comprehensive theory to account for trait-performance relations (Matthews et al., 2003). Most current theories are more like frameworks that do not provide specific mechanisms for how personality impacts cognition and performance (e.g., see McCrae & Costa, 1999). Although H. J. Eysenck (1967) proposed a theory of personality based on arousal and activation, which has found some support (Eysenck & Eysenck, 1985), there has also been evidence that arousal and task difficulty fail to interact as predicted (Matthews, 1992). H. J. Eysenck's (1967) theory was also weakened by the general problems associated with arousal theory accounts for stress effects (Hockey, R., 1984). An alternative formulation is that of Gray (1991), who argued for two systems, one responding to reward signals and one with

punishment. The behavioral activation system (BAS) is associated with positive affect, while the behavioral inhibition system with negative affect. In a review and comparisons of the H. J. Eysenck and Gray theories, Matthews and Gilliland (1999) concluded that both theories have only been partially supported, but that Gray's BAS/BIS distinction provides a superior match to positive and negative affect relative to H. J. Eysenck's arousal dimensions. Further, the BAS/BIS accords with theories of approach/avoidance motivation (e.g., Elliot & Covington, 2001). There are also theories that focus on a particular trait, such as Extraversion (Humphreys & Revelle, 1984) or trait anxiety (Eysenck, M. W., & Calvo, 1992). While useful, such specific theories do not encompass other traits or interactions among traits. Such interactive effects can influence cognitive performance and perceived stress and workload (Szalma, Oron-Gilad, Stafford, & Hancock, 2005). These interactions should be further studied with an eye to linking them to information processing theories.

Affective State Differences

It is intuitive that stress would induce more negative affective states, and that traits would influence performance via an effect on states. For instance, one would expect that trait anxiety would influence performance because high trait anxious individuals experience state anxiety more frequently than those low on that trait. While such mediation effects are observed, there is also evidence that, for certain processes, such as hyper vigilance to threat, trait anxiety is a better predictor of performance than state anxiety (Eysenck, M. W., 1992). In terms of appraisal theory, traits may influence the form and content of appraisal, as well as the coping skills the individual can deploy to deal with the stress. In regard to the adaptation, it is likely that individual differences in both trait and state will influence adaptation, both behavioral and physiological, by affecting the width of the plateau of effective adaptation at a given level, and by changing the slope of decline in adaptation when the adaptation threshold has been reached. That is, higher skill levels protect from declines in adaptive function by increasing the threshold for failure at a given level (e.g., comfort, performance, physiological response). The modification of the Hancock and Warm (1989) model, illustrating these individual differences effects, is shown in Fig. 6.6. Multiple frameworks of state dimensions exist, but most focus on either two (e.g., Thayer, 1989; Watson & Tellegen, 1985), or three (Matthews et al., 1999, 2002). In the context of task performance, Matthews and his colleagues identified three

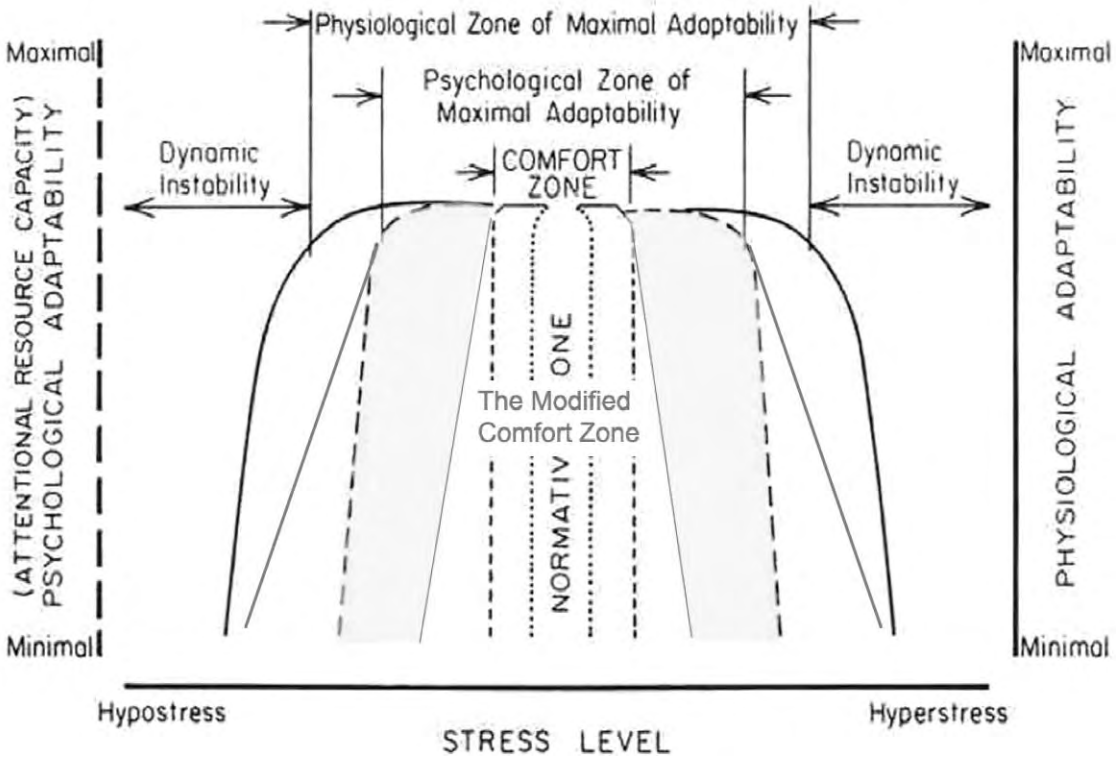


FIGURE 6.6. The adaptability model of Hancock and Warm (1989) shown in Figure 3 has been modified to illustrate how individual differences may influence stress and adaptation. It is likely that cognitive and affective traits influence both the width of the comfort and performance zones (i.e., the 'thresholds' for declines in adaptation) as well as the rate of decline in adaptability when a threshold has been crossed. For instance, individuals high in trait anxiety would likely have a narrower plateau of stability and would therefore manifest lower thresholds for discomfort and performance degradation than individuals low on that trait. Further, the rate of decline in adaptation may increase as a function of trait anxiety.

broad state dimensions reflecting the cognitive, affective, and motivational aspects of an individual's current psychological state. These dimensions are "worry," which reflects the cognitive dimension of stress, and "task engagement" and "distress," which reflect the affective, cognitive, and motivational components of state. Specifically, a high level of distress is indicative of overload in processing capacity, while task engagement reflects a theme of commitment to effort (Matthews et al., 2002). Matthews and his colleagues (2002) demonstrated that changes in task demand influence the pattern of stress state. It is therefore important to incorporate assessment of operator state into the interface design process so that the interaction with the technology fosters task engagement and minimizes distress and worry.

Attentional narrowing and adaptive response. As with other aspects of perception, there are individual differences in the perception of space and time (Hancock & Weaver, 2002; Wachtel, 1967). Further, because the subjective experience of stress is often multidimensional, it may be that two individuals are subjectively stressed by the same situation but that their stress state profiles differ. Individuals are also likely to differ in the strategies they employ to cope with the distortions of space-time they experience while in a stressful environment, and these coping differences, if they exist, might depend on the quality (e.g. noise, heat, low signal salience) and source (e.g. environment, the task) of the stress and the personality traits of the individual.

Hedonomics and individual differences. In addition to application of individual differences research to development of training or selection procedures, individual of relevant individual different variables can promote hedonomic approaches to design and facilitate individuation in interface design. Thus, if the traits that influence the subjective experience of an interaction with technology are identified, that interface can then be configured to meet the preferences and the trait/state profile of the individual user and promote positive affective states. However, for such efforts to succeed, the relations among traits and cognitive, perceptual, and motor performance will need to be established via theory-guided empirical research.

IMPLICATIONS OF STRESS FOR RESEARCHERS AND PRACTITIONERS

For both research and design application, the extant research on stress and performance indicates that assessment of workload and affective state are important for a more complete understanding of HCI. Such assessments can aid in identifying which components of an interface or task are appraised as stressful and thereby design to mitigate their negative effects. For instance, research is needed to establish which task parameters control the patterns of performance-workload associations and dissociations, and how these change dynamically as a function of time on task. The Hancock and Warm (1989) model of stress established general task dimensions (space-time) that influence stress state and behavioral adaptability, but the metrics for these dimensions remain elusive. This problem results from the central issue regarding how to quantify human information processing (Hancock, Szalma, & Oron-Gilad, 2005) and define mental resources more precisely (Hancock & Szalma, 2007). Efforts to resolve these definitional problems would improve stress theory and its application to interface design. Future research should therefore examine the relations between task dimensions and user characteristics, and how these change over time and under high-stress conditions.

In addition to changing the task, there are other techniques that can be applied to the design of HCIs for use in stressful environments. These include skill development (e.g., Hancock, 1986) and use of configural displays (Hancock & Szalma, 2003a; Wickens, 1996), as well as technologies employing adaptive automation and decision aids (Hancock & Chignell, 1987). In regard to skill development in particular, an area in need of research is how individuals who develop expertise also learn how to cope with stress while performing the task. Understanding how individuals accomplish this will require advances in understanding how different forms of stress influence different forms of information processing.

It is also important for both researchers and practitioners to consider the characteristics of the user and to consider how these characteristics interact with the task or interface to influence performance. Understanding how individual differences influence human-computer interaction can facilitate development of tailored training regimens as well as interfaces that more effectively adapt to the user. Systems that can respond to changes in operator affective state will achieve the desired human-machine synergy in HCI (c.f., Hancock, 1997). Realizing these goals, however, will require adequate theory development and subsequent empirical research to determine the nature of the relations among the person and environmental variables. It will be particularly important to design interfaces that permit autonomous motivation (Deci & Ryan, 2000), and to understand how operators of computer-based systems can internalize extrinsic motivation as they gain experience with the task. (Gagne & Deci, 2005). We suggest here that researchers and designers identify the motivational affordances in the task environment and utilize these to enhance the experience of HCI and improve overall system performance under stress. Motivational affordances will be elements of the work environment that facilitate and nurture intrinsic motivation. Particularly important for design will be to identify motivational invariants, which are those environmental factors that consistently determine an individual's level of intrinsic (or extrinsic) motivation across contexts. Careful analyses of the motivational affordance structure will permit design of tasks that are more likely to be enjoyable by rendering the tools convivial (Illich, 1973) and thereby facilitating human-machine synergy (and see Hancock, 1997).

SUMMARY AND CONCLUSIONS

In this chapter, we reviewed three theories of stress and performance and their relevance for human-technology interaction. We also showed that despite separate research traditions, work-

load and stress might be viewed as different perspectives on the same problem. We outlined some general principles for stress mitigation, and issues requiring further research. Of particular importance will be establishing sound measures of information processing and mental resources, as well as articulating the relevant task dimensions and how they relate to self-regulatory mechanisms. Given that stress can only be understood in relation to the transaction between an individual and the environment, it will be crucial to establish how traits and states of the individual influence their appraisals of their environments. Finally, it will be important in practical application to treat stress at multiple levels, ranging from the physiological to the organizational sources of adverse performance effects. Traditional attempts to treat stress problems unidimensionally will continue to fail until the person, task, and physical, social, and organizational environments are treated as a system. Researchers and practitioners in HCI should therefore expand their efforts beyond the design of the displays and controls of interfaces and include assessment of the person factors that influence performance as well as the design of the physical and social environment in which the human-computer interaction occurs.

The argument here is not that neural structures are not constrained by the laws of thermodynamics—clearly they are—but that thermodynamic principles implied by the metaphor are not sufficient for the development of a complete description of resources and their relation to cognitive activity.

References

Ackerman, P. L., & Heggestad, E. D. (1997). Intelligence, personality, and interests: Evidence for overlapping traits. *Psychological Bulletin, 121*, 219–245.

Aspinwall, L. G., Richter, L., & Hoffman, R. R. (2002). Understanding how optimism works: An examination of optimists' adaptive moderation of belief and behavior. In E. C. Chang (Ed.), *Optimism and pessimism: Implications for theory, research, and practice* (pp. 217–238). Washington, DC: American Psychological Association.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. NY: W.H. Freeman and Company.

Barrick, M. R., Mount, M. K., & Judge, T. A. (2001). Personality and performance at the beginning of the new millennium: What do we know and where do we go next? *Personality and Performance, 9*, 9–29.

Bennett, K. B., & Flach, J. M. (1992). Graphical displays: Implications for divided attention, focused attention, and problem solving. *Human Factors, 34*, 513–552.

Bless, H. (2001). Mood and the use of general knowledge structures. In L. L. Martin & G. L. Clore (Eds.), *Theories of mood and cognition: A user's guidebook* (pp. 9–26). Mahwah, NJ: Erlbaum.

Brown, J. M., & Campbell, E. A. (1994). *Stress and policing: Sources and strategies.* Chichester, UK: Wiley.

Bursill, A. E. (1958). The restriction of peripheral vision during exposure to hot and humid conditions. *Quarterly Journal of Experimental Psychology, 10*, 113–129.

Carver, C. S., & Scheier, M. F. (1998). *On the self-regulation of behavior*. NY: Cambridge University Press.

Cornsweet, D. M. (1969). Use of cues in the visual periphery under conditions of arousal. *Journal of Experimental Psychology, 80*, 14–18.

Cronbach, L. J. (1957). The two disciplines of scientific psychology. *American Psychologist, 12*, 671–684.

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. NY: Harper & Row.

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. NY: Plenum Press.

Deci, E. L., & Ryan, R. M. (2000). The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological Inquiry, 11*, 227–268.

Deci, E. L., Ryan, R. M., & Koestner, R. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin, 125*, 627–668.

Dirkin, G. R., & Hancock, P. A. (1984). Attentional narrowing to the visual periphery under temporal and acoustic stress. *Aviation, Space and Environmental Medicine, 55*, 457.

Dirkin, G. R., & Hancock, P. A. (1985). An attentional view of narrowing: The effect of noise and signal bias on discrimination in the peripheral visual field. In I. D. Brown, R. Goldsmith, K. Coombes, & M. A. Sinclair (Eds.), *Ergonomics International 85: Proceedings of the Ninth Congress of the International Ergonomics Association*, (pp. 751–753). Bournemouth, England, September.

Driskell, J. E., Johnston, J. H., & Salas, E. (2001). Does stress training generalize to novel settings? *Human Factors, 43*, 99–110.

Easterbrook, J. A. (1959). The effect of emotion on cue utilization and the organization of behavior. *Psychological Review, 66*, 183–201.

Elliot, A. J., & Covington, M. V. (2001). Approach and avoidance motivation. *Educational Psychology Review, 13*, 73–92.

Ericsson, K. A. (2006). The influence of experience and deliberate practice on the development of superior expert performance. In K. A. Ericsson, N. Charness, R. R. Hoffman, & P. J. Feltovich (Eds.), *The Cambridge handbook of expertise and expert performance (*pp. 683–703). Cambridge: Cambridge University Press.

Eysenck, H. J. (1967). *The biological basis of personality*. Springfield, IL: Charles C. Thomas.

Eysenck, H. J., & Eysenck, M. W. (1985). *Personality and individual differences: A natural science approach.* New York: Plenum Press.

Eysenck, M. W. (1992). *Anxiety: The cognitive perspective*. Hillsdale, NJ: Erlbaum.

Eysenck, M. W., & Calvo, M. G. (1992). Anxiety and performance: The processing efficiency theory. *Cognition and Emotion, 6*, 409–434.

Flach, J., Hancock, P., Caird, J., & Vicente, K. (Eds.). (1995). *Global perspectives on the ecology of human-machine systems vol 1*. Hillsdale, NJ: Erlbaum.

Fromm, E. (1976). *To have or to be?* NY: Harper & Row.

Gagne, M., & Deci, E. L. (2005). Self-determination theory and work motivation. *Journal of Organizational Behavior, 26*, 331–362.

Gibson, J. J. (1966). *The senses considered as perceptual systems*. Boston: Houghton Mifflin.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston: Houghton Mifflin.

Gray, J. A. (1991). Neural systems, emotion, and personality. In J. Madden, IV (Ed.), *Neurobiology of learning, emotion, and affect*. (pp. 273–306). NY: Raven Press.

Hancock, P. A. (1986). The effect of skill on performance under an environmental stressor. *Aviation, Space, and Environmental Medicine, 57*, 59–64.

Hancock, P. A. (1996). Effects of control order, augmented feedback, input device and practice on tracking performance and perceived workload. *Ergonomics, 39*, 1146–1162.

Hancock, P. A. (1997). *Essays on the future of human-machine systems*. Eden Prairie, MN: BANTA Information Services Group.

Hancock, P. A., & Caird, J. K. (1993). Experimental evaluation of a model of mental workload. *Human Factors, 35*, 413–429.

Hancock, P. A., & Chignell, M. H. (1987). Adaptive control in human-machine systems. In P. A. Hancock (Ed.), *Human factors psychology* (pp. 305–345). Amsterdam: Elsevier.

Hancock, P. A., & Dirkin, G. R. (1983). Stressor induced attentional narrowing: Implications for design and operation of person-machine systems. *Proceedings of the Human Factors Association of Canada, 16*, 19–21.

Hancock, P. A., Pepe, A. A., & Murphy, L. L. (2005). Hedonomics: The power of positive and pleasurable ergonomics. *Ergonomics in Design, 13*, 8–14.

Hancock, P. A., Ross, J. M., & Szalma, J. L. (in press). A meta-analysis of performance response under thermal stressors. *Human Factors.*

Hancock, P. A., & Szalma, J. L. (2003a). Operator stress and display design. *Ergonomics in Design, 11*, 13–18.

Hancock, P. A., & Szalma, J. L. (2003b) The future of neuroergonomics. *Theoretical Issues in Ergonomics Science, 4*, 238–249.

Hancock, P. A., & Szalma, J. L. (2007). Stress and neuroergonomics. In R. Parasuraman & M. Rizzo (Eds.), *Neuroergonomics: The brain at work* (pp. 195–206). Oxford: Oxford University Press.

Hancock, P. A., Szalma, J. L., & Oron-Gilad, T. (2005). Time, emotion, an the limits to human information processing. In D. K. McBride & D. Schmorrow (Eds.), *Quantifying Human Information Processing* (pp. 157–175). Lanham, MD: Lexington Books.

Hancock, P. A., & Warm, J. S. (1989). A dynamic model of stress and sustained attention. *Human Factors, 31*, 519–537.

Hancock, P. A., & Weaver, J. L. (2005). On time distortion under stress. *Theoretical Issues in Ergonomics Science, 6*, 193–211.

Harris, W. C., Hancock, P. A., & Harris, S. C. (2005) Information processing changes following extended stress. *Military Psychology, 17*, 115–128.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–183). Amsterdam: Elsevier.

Hockey, G. R. J. (1997). Compensatory control in the regulation of human performance under stress and high workload: A cognitive-energetical framework. *Biological Psychology, 45*, 73–93.

Hockey, G. R. J., Gaillard, A. W. K., & Coles, M. G. H. (Eds.). (1986). *Energetic aspects of human information processing*. The Netherlands: Nijhoff.

Hockey, R. (1984). Varieties of attentional state: The effects of environment. In R. Parasuraman & D. R. Davies (Eds.), *Varieties of attention* (pp. 449–483). NY: Academic Press.

Hockey, R., & Hamilton, P. (1983). The cognitive patterning of stress states. In G. R. J. Hockey (Ed.), *Stress and fatigue in human performance* (pp. 331–362). Chichester: Wiley.

Humphreys, M. S., & Revelle, W. (1984). Personality, motivation, and performance: A theory of the relationship between individual differences and information processing. *Psychological Review, 91*, 153–184.

Illich, I. (1973). *Tools for conviviality*. NY: Harper & Row.

Johnston, J. H., & Cannon-Bowers, J. A. (1996). Training for stress exposure. In J. E. Driskell & E. Salas (Eds.), *Stress and human performance.* (pp. 223–256) Mahwah, NJ: Erlbaum.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice Hall.

Kasser, T. (2002). The *high price of materialism*. Cambridge, MA: MIT Press.

Koelega, H. S. (1992). Extraversion and vigilance performance: Thirty years of inconsistencies. *Psychological Bulletin, 112*, 239–258.

Lazarus, R. S. (1991). *Emotion and adaptation*. Oxford: Oxford University Press.

Lazarus, R. S. (1999). *Stress and emotion: A new synthesis*. NY: Springer.

Lazarus, R. S., & Folkman, S. (1984). *Stress, appraisal, and coping*. NY: Springer Verlag.

Luczak, H., Roettling, M., & Schmidt, L. (2003). Let's talk: Anthropomorphization as means to cope with stress of interacting with technical devices. *Ergonomics, 46*, 1361–1374.

McCrae, R. R., & Costa, P. T. (1986). Personality, coping, and coping effectiveness in an adult sample. *Journal of Personality, 54*, 385–405.

McCrae, R. R., & Costa, P. T. (1999). A five-factor theory of personality. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research, 2nd ed.* (pp. 139–153). NY: Guilford Press.

Matthews, G. (1992). Extraversion. In A. P. Smith & D. M. Jones (Eds.), *Handbook of human performance, Vol. 3: State and Trait* (pp. 95–126). London: Academic Press.

Matthews, G. (2001). Levels of transaction: A cognitive science framework for operator stress. In P. A. Hancock & P. A. Desmond (Eds.), *Stress, workload, and fatigue* (pp. 5–33). Mahwah, NJ: Erlbaum.

Matthews, G., Deary, I. J., & Whiteman, M. C. (2003). *Personality traits, 2nd ed*. Cambridge: Cambridge University Press.

Matthews, G., & Gilliland, K. (1999). The personality theories of H. J. Eysenck and J. A. Gray: A comparative review. *Personality and Individual Differences, 26*, 583–626.

Matthews, G., Joyner, L., Gilliland, K., Campbell, S., Falconer, S., & Huggins, J. (1999). Validation of a comprehensive stress state questionnaire: Towards a state 'big three'? In I. Mervielde, I. J. Deary, F. De Fruyt, & F. Ostendorf (Eds.), *Personality Psychology in Europe: Vol. 7* (pp. 335–350). Tilburg, the Netherlands: Tilburg University Press.

Matthews, G., Campbell, S. E., Falconer, S., Joyner, L. A., Huggins, J., Gilliland, et al. (2002). Fundamental dimensions of subjective state in performance settings: Task engagement, distress, and worry. *Emotion, 2*, 315–340.

Moray, N. (1967). Where is capacity limited? A survey and a model. *Acta Psychologica, 27*, 84–92.

Moray, N. (Ed.). (1979). *Mental workload: Its theory and measurement*. NY: Plenum Press.

Navon, D. (1984). Resources—A theoretical soupstone? *Psychological Review, 91*, 216–234.

Navon, D., & Gopher, D. (1979). On the economy of the human information processing system. *Psychological Review, 86*, 214–255.

Norman, D., & Bobrow, D. (1975). On data-limited and resource-limited processing. *Journal of Cognitive Psychology, 7*, 44–60.

O'Donnell, R. D., & Eggemeier, F. T. (1986). Workload assessment methodology. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of human perception and performance, Vol. II: Cognitive processes and performance.* (pp. 42-1–42-49). NY: Wiley.

Parasuraman, R. (2003). Neuroergonomics: Research and practice. *Theoretical Issues in Ergonomics Science, 4*, 5–20.

Parasuraman, R., & Hancock, P. A. (2001). Adaptive control of mental workload. In P. A. Hancock & P. A. Desmond (Eds.), *Stress, workload, and fatigue.* (pp. 305–320). Mahwah, NJ: Erlbaum.

Penley, J. A., & Tomaka, J. (2002). Associations among the big five, emotional responses, and coping with acute stress. *Personality and Individual Differences, 32*, 1215–1228.

Pilcher, J. J., Nadler, E., & Busch, C. (2002). Effects of hot and cold temperature exposure on performance: A meta-analytic review. *Ergonomics, 45,* 682–698.

Rasmussen, J. (1983). Skills, rules, and knowledge; Signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-13*, 257–266.

Reed, E. S. (1996). *Encountering the world: Toward an ecological psychology*. Oxford: Oxford University Press.

Ross, J. M., Szalma, J. L., Thropp, J. E., & Hancock, P. A. (2003). Performance, workload, and stress correlates of temporal and spatial task demands. *Proceedings of the Human Factors and Ergonomics Society, 47*, 1712–1716.

Rugg, M. D. (1986). Constraints on cognitive performance: Some problems with and alternatives to resource theory. In G. R. J. Hockey,

A. W. K. Gaillard, & M. G. H. Coles (Eds.), *Energetics and human information processing* (pp. 353–371). Dordrecht: Martinus Nijhoff Publishers.

Ryan, R. R., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist, 55*, 66–78.

Ryan, R. R., & Deci, E. L. (2001). On happiness and human potentials: A review of research on hedonic and eudaimonic well-being. *Annual Review of Psychology, 52*, 141–166.

Saunders, T., Driskell, J. E., Johnston, J., & Salas, E. (1996). The effect of stress inoculation training on anxiety and performance. *Journal of Occupational Health Psychology, 1,* 170–186.

Scerbo, M. W. (2007). Adaptive automation. In: R. Parasuraman and M. Rizzo (Eds.). *Neuroergonomcs: The brain at work* (pp. 239–252). Oxford: Oxford University Press.

Scerbo, M. W., Freeman, F. G., & Mikulka, P. J. (2003). A brain-based system for adaptive automation. *Theoretical Issues in Ergonomics Science, 4*, 200–219.

Scheier, M. F., & Carver, C. S. (1985). Optimism, coping, and health: Assessment and implications of generalized outcome expectancies. *Health Psychology, 4*, 219–247.

Scherer, K. R. (1999). Appraisal theory. In T. Dalgleish & M. Power (Eds.), *Handbook of cognition and emotion* (pp. 638–663). NY: Wiley.

Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing I: Detection, search, and attention. *Psychological Review, 84*, 1–66.

Seligman, M. E. P., & Csikszentmihalyi, M. (2000). Positive psychology: An introduction. *American Psychologist, 55*, 5–14.

Selye, H. (1976). *The stress of life, revised edition*. NY: McGraw-Hill.

Szalma, J. L. (1999). *Sensory and temporal determinants of workload and stress in sustained attention*. Unpublished doctoral dissertation, University of Cincinnati: Cincinnati, OH.

Szalma, J. L., Warm, J. S., Matthews, G., Dember, W. N., Weiler, E. M., Meier, A., et al. (2004). Effects of sensory modality and task duration on performance, workload, and stress in sustained attention. *Human Factors, 46*, 219–233.

Szalma, J. L., & Hancock, P. A. (2005). Individual differences in information processing. In D. K. McBride & D. Schmorrow (Eds.), *Quantifying human information processing* (pp. 177–193). Lanham, MD: Lexington Books.

Szalma, Oron-Gilad, Stafford, & Hancock (2007). Police firearms training: Individual differences in performance, workload, and stress.

Thayer, R. E. (1989). *The biopsychology of mood and arousal*. NY: Oxford University Press.

Thropp, J. E., Oron-Gilad, Szalma, & Hancock (2007). *Individual preferences using automation.* Manuscript in preparation.

Thropp, J. E., Szalma, J. L., & Hancock, P. A. (2004). Performance operating characteristics for spatial and temporal discriminations: Common or separate capacities? *Proceedings of the Human Factors and Ergonomics Society, 48*, 1880–1884.

van Reekum, C., & Scherer, K. R. (1997). Levels of processing in emotion-antecedent appraisal. In G. Matthews (Ed.), *Cognitive science perspectives on personality and emotio*n (pp. 259–300). Amsterdam: Elsevier.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics, 22*, 589–606.

Wachtel, P. L. (1967). Conceptions of broad and narrow attention. *Psychological Bulletin, 68*, 417–429.

Warm, J. S. (1993). Vigilance and target detection. In B. M. Huey & C. D. Wickens (Eds.), *Workload transition: Implications for individual and team performance* (pp. 139–170). Washington, DC: National Academy Press.

Warm, J. S., Dember, W. N., & Hancock, P. A. (1996). Vigilance and workload in automated systems. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp. 183–200). Hillsdale, NJ: Erlbaum.

Watson, D., & Tellegen, A. (1985). Toward a consensual structure of mood. *Psychological Bulletin, 98,* 219–235.

Wickens, C. D. (1980). The structure of attentional resources. In R. Nickerson (Ed.), *Attention and performance VIII* (pp. 239–257). Hillsdale, NJ: Erlbaum.

Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman & D. R. Davies (Eds.), *Varieties of attention* (pp. 63–102). New York, NY: Academic Press.

Wickens, C. D. (1996). Designing for stress. In J. E. Driskell & E. Salas (Eds.), *Stress and human performance* (pp. 279–295). Mahwah, NJ: Erlbaum.

Yeh, Y., & Wickens, C. D. (1988). Dissociation of performance and subjective measures of workload. *Human Factors, 30*, 111–120.

Yerkes, R. (1918). Psychology in relation to the war. *Psychological Review, 25*, 85–115.

Zimmerman, B. J. (2000). Attaining self-regulation: A social cognitive perspective. In M. Boekaerts, P. R. Pintrch, & M. Zeidner (Eds.), *Handbook of self-regulatio*n (pp. 13–39). San Diego: Academic Press.

◆7◆

MOTIVATING, INFLUENCING, AND PERSUADING USERS: AN INTRODUCTION TO CAPTOLOGY

B.J. Fogg,* Gregory Cuellar, David Danielson
Stanford University

Defining Persuasion and Giving High-Tech Examples **134**
The Fourth Wave: Persuasive Interactive Technology **135**
Domains Where Persuasion and Motivation Matters **135**
Persuasion and Interactive Technology: Two Levels of Analysis **136**
Examples of Macrosuasion 136
Examples of Microsuasion 136
No Universal Theory of Persuasion **137**
The Functional Triad: A Framework for Persuasive Technology **137**
Computers as Persuasive Tools 137
Computers That Increase Self-Efficacy 137
Computers That Provide Tailored Information 138
Computers That Trigger Decision-Making 138
Computers That Simplify or Guide People Through a Process 138
Computers as Persuasive Media 138
Computers That Simulate Cause and Effect 138
Computers That Simulate Environments 139
Computers That Simulate Objects 139
Computers as Persuasive Social Actors 139
Computers That Provide Social Support 139
Computers That Model Attitudes and Behaviors . . . 140
Computers That Leverage Social Rules and Dynamics 140
Functional Triad Summary 140
Computers and Credibility **140**
When Does Credibility Matter? 141
1. When computers act as a knowledge repository 141
2. When computers instruct or tutor users 141
3. When computers report measurements 141
4. When computers report on work performed 141
5. When computers report about their own state 141
6. When computers run simulations 141
7. When computers render virtual environments 142
Web Credibility Research and Guidelines for Design 142
Guideline #1: Design Websites to convey the "real world" aspect of the organization 142
Guideline #2: Invest resources in visual design 142

*B.J. Fogg directs the Stanford Persuasive Technology Lab and is on the consulting faculty for Stanford's Computer Science Department.

Guideline #3: Make Websites easy to use 143
Guideline #4: Include markers of expertise 143
Guideline #5: Include markers of trustworthiness 143
Guideline #6: Tailor the user experience 143
Guideline #7. Avoid overly commercial elements on a Website 143
Guideline #8. Avoid the pitfalls of amateurism . . . 143
Positive and Negative Applications of Persuasive Technology 143
The Ethics of Computing Systems Designed to Persuade 144
Persuasive Technology: Potential and Responsibility 144
References 144

Since the advent of modern computing in 1946, the uses of computing technology have expanded far beyond their initial role of performing complex calculations (Denning & Metcalfe, 1997). Computers are not just for scientists any more; they are an integral part of workplaces and homes. The diffusion of computers has led to new uses for interactive technology; including the use of computers to change people's attitudes and behavior—in a word: persuasion. Computing pioneers of the 1940s probably never imagined computers being used to persuade.

Today, creating successful human-computer interactions (HCIs) requires skills in motivating and persuading people. However, interaction designers don't often view themselves as agents of influence. They should. The work they perform often includes crafting experiences that change people—the way people feel, what they believe, and the way in which they behave. Consider these common challenges: How can designers motivate people to register their software? How can they get people to persist in learning an online application? How can they create experiences that build product loyalty? Often, the success of today's interactive product hinges on changing people's attitudes or behaviors.

Sometimes the influence elements in HCI are small, almost imperceptible, such as creating a feeling of confidence or trust in what the computing product says or does. Other times, the influence element is large, even life altering, such as motivating someone to quit smoking. Small or large, elements of influence are increasingly present on Websites, in productivity tools, in video games, in wearable devices, and in other types of interactive computing products. Due to the growing use of computing products and to the unparalleled ability of software to scale, interaction designers may well become leading change agents of the future. Are we ready?

The study and design of computers as persuasive technologies, referred to as *captology*, is a relatively new endeavor when compared to other areas of HCI (Fogg, 1997, 1998, 2003). Fortunately, understanding in this area is growing. HCI professionals have established a foundation that outlines the domains of applications, useful frameworks, methods for research, design guidelines, best-in-class examples, as well as ethical issues (Berdichevsky & Neuenschwander, 1999; Fogg, 1999; Khaslavsky & Shedroff, 1999; King & Tester, 1999; Tseng & Fogg, 1999). This chapter will not address all these areas in-depth, but it will share some key perspectives, frameworks, and design guidelines relating to captology.

DEFINING PERSUASION AND GIVING HIGH-TECH EXAMPLES

What is "persuasion"? As one might predict, scholars do not agree on the precise definition. For the sake of this chapter, *persuasion is a noncoercive attempt to change attitudes or behaviors.* There are some important things to note about this definition. First, persuasion is *noncoercive*. Coercion—the use of force—is not persuasion; neither is manipulation or deceit. These methods are shortcuts to changing how people believe or behave, and for interaction designers these methods are rarely justifiable.

Next, persuasion requires an *attempt* to change another person. The word *attempt* implies intentionality. If a person changes someone else's attitude or behavior without intent to do so, it is an accident or a side effect; it is not persuasion. This point about intentionality may seem subtle, but it is not trivial. Intentionality distinguishes between a *side effect* and a *planned effect* of a technology. At its essence, captology focuses on the planned persuasive effects of computer technologies.

Finally, persuasion deals with *attitude changes* or *behavior changes* or both. While some scholars contend persuasion pertains only to attitude change, other scholars would concur with our view: including behavior change as a target outcome of persuasion. Indeed, these two outcomes—attitude change and behavior change—are fundamental in the study of computers as persuasive technologies.

Note how attitude and behavior changes are central in two examples of persuasive technology products. First, consider the CD-ROM product *5 A Day Adventures* (www.dole5aday.com). Created by Dole Foods, this computer application was designed to persuade kids to eat more fruits and vegetables. Using *5 A Day Adventures*, children enter a virtual world with characters like "Bobby Banana" and "Pamela Pineapple," who teach kids about nutrition and coach them to make healthy food choices. The program also offers areas where children can practice making meals using fresh produce, and the virtual characters offer feedback and praise. This product clearly aims to change the attitudes children have about eating fruits and vegetables. However, even more important, the product sets out to change their eating behaviors.

Next, consider a more mundane example: Amazon.com. The goal of this Website is to persuade people to buy products again and again from Amazon.com. Everything on the Website con-

tributes to this result: user registration, tailored information, limited-time offers, third-party product reviews, one-click shopping, confirmation messages, and more. Dozens of persuasion strategies are integrated into the overall experience. Although the Amazon online experience may appear to be focused on providing mere information and seamless service, it is really about persuasion—buy things now and come back for more.

THE FOURTH WAVE: PERSUASIVE INTERACTIVE TECHNOLOGY

Computing systems did not always contain elements of influence. It has only been in recent years that interactive computing became mature enough to spawn applications with explicit elements of influence. The dramatic growth of technologies designed to persuade and motivate represents the fourth wave of focus in end-user computing. The fourth wave leverages advances from the three previous waves (Fig. 7.1).

The first wave of computing began over 50 years ago and continues today. The energy and attention of computer professionals mainly focused on getting computing devices to work properly, and then to make them more and more capable. In short, the first wave is function.

The second wave of computing began in the 1970s with the emergence of digital gaming, first represented by companies like Atari and with products like Pong. This wave is entertainment, and it continues to swell because of continued attention and energy devoted to computer-based fun.

The third wave of computing came in the 1980s when human factors specialists, designers, and psychologists sought to create computers for ordinary people. This third wave is ease of use. Although new developments, like the computer mouse and the graphical-user interface came before 1980, a consumer product—the Apple Macintosh— generated widespread attention and energy to making computers easier to use. Like the previous two waves, the third wave keeps rolling today. It provides the foundation for most work in HCI arenas.

In addition, this brings us to the fourth wave: computers designed to persuade. Early signs of this wave appeared in the 1970s and 1980s with a handful of computing systems designed to motivate health behaviors and work productivity. However, it wasn't until the late-1990s—specifically during the rise of the World Wide Web—that more than a handful of people began to devote attention and energy to making interactive systems capable of motivating and influencing users. This fourth wave—persuasion—is new and could be as significant as the three waves that have come before it.

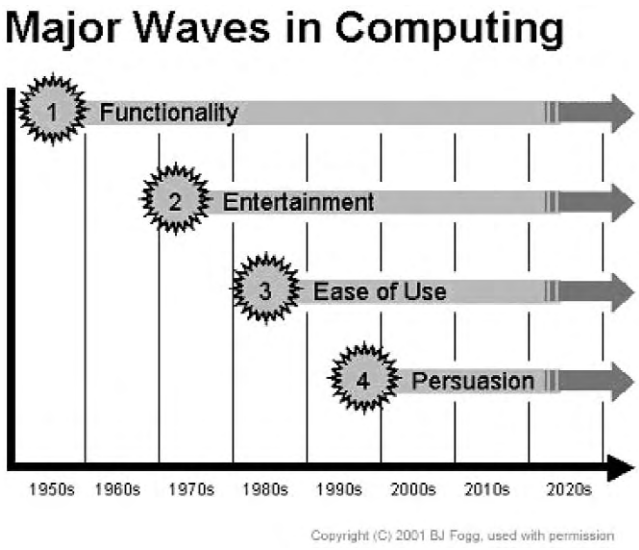


FIGURE 7.1. Persuasion is the fourth major wave of computing.

DOMAINS WHERE PERSUASION AND MOTIVATION MATTER

Captology is relevant to systems designed for many facets of human life. The most obvious domain is in promoting commerce—buying and branding, especially via the Web. While promoting commerce is perhaps the most obvious and lucrative application, at least 11 other domains are potential areas for persuasive technology products. The various domains, along with a sample target behavior change, are summarized in Table 7.1.

The domains in the table reflect how much persuasion is part of ordinary human experience, from personal relationships to environmental conservation. Interactive technologies have been—and will continue to be—created to influence people in these 12 domains, as well as in others that are less apparent. The way various computing products incorporate persuasion and motivation principles will evolve as computing technology ad-

TABLE 7.1. 12 Domains for Persuasive Technology

Domains for Persuasive Technologies	Example
Commerce—Buying and Branding	To buy a certain product
Education, Learning, and Training	To engage in activities that promote learning
Safety	To drive more safely
Environmental Conservation	To reuse shopping bags
Occupational Productivity	To set and achieve goals at work
Preventative Health Care	To quit smoking
Fitness	To exercise with optimal intensity/frequency
Disease Management	To manage diabetes better
Personal Finance	To create and adhere to a personal budget
Community Involvement/Activism	To volunteer time at a community center
Personal Relationships	To keep in touch with their aging parents
Personal Management and Improvement	To avoid procrastination

vances and as people adopt a wider array of interactive systems for a wider range of human activities. The influence elements in these systems can be readily apparent, or they can be woven into the fabric of an interactive experience, a distinction explored in the next section.

PERSUASION AND INTERACTIVE TECHNOLOGY: TWO LEVELS OF ANALYSIS

One key insight in captology is to see that persuasion in computing products takes place on two levels: macro and micro. On the macro level, one finds products designed for an overall persuasive outcome. For example, the Dole *5 A Day* CD-ROM and the Amazon.com Website are designed specifically for persuasion. For these and other products, persuasion and motivation are the sole reason these products exist. We use the word "macrosuasion" to describe this type of big-picture target outcome.

On the other hand, one finds computing products with what I call "microsuasion." These products could be word-processing programs or spreadsheets; they do not necessarily have a persuasive outcome as the overall goal of the product. However, they will incorporate smaller elements of influence to achieve other goals. Microsuasion can be incorporated into dialogue boxes, visual elements, interactions sequences, and more. In productivity software, microsuasion can lead to increased productivity or stronger brand loyalty. The following examples will help clarify the distinction between macrosuasion and microsuasion (Fogg, 2003).

Examples of Macrosuasion

One notable example of macrosuasion is a product named Baby Think It Over. A U.S. company (www.btio.com) designed this computerized doll to simulate the time and energy required to care for a baby, with the overall purpose of persuading teens to avoid becoming parents prematurely. Used as part of many school programs in the United States, the Baby Think It Over infant simulator looks, weighs, and cries something like a real baby. The computer embedded inside the doll triggers a crying sound at random intervals; in order to stop the crying sound, the teen caregiver must pay immediate attention to the doll. If the caregiver fails to respond appropriately, the computed embedded inside the doll records the neglect. After a few days of caring for the simulated infant, teenagers generally report less interest in becoming a parent in the near future (see www.btio.com), which—along with reduced teen pregnancy rates—is the intended outcome of the device.

Next, consider Scorecard.org as another example of macrosuasion. Created by the Environmental Defense Foundation, this Website helps people find information about pollution threats in their neighborhoods. When users enter their zip code, the site lists names of the polluting institutions in their area, gives data on chemicals being released, and outlines the possible health consequences. But that's not all. Scorecard.org then encourages users to take action against the polluting organizations and makes it easy to contact policymakers to express concerns. This Website aims to increase community activism in order to pressure officials and offending institutions into cleaning up the environment. The entire point of this Website is to get people to take action against polluting institutions in their neighborhoods. This is macrosuasion.

Examples of Microsuasion

Most computing products were not created with persuasion as the main focus. Larger software categories include applications for productivity, entertainment, and creativity. Yet these same products often use influence elements, microsuasion, as part of the overall experience. Examples of interactive products using microsuasion are plentiful—and sometimes subtle. A word-processing program may encourage users to spell check text, or a Website devoted to high-chool reunions may reward alumni for posting a current photograph online. This is persuasion on a microlevel.

For a deeper look at microsuasion, consider the personal finance application Quicken, created by Intuit (www.intuit.com). Quicken is a productivity software product. Its overall goal is to simplify the process of managing personal finances. Quicken uses microsuasion to accomplish this overall goal. For example, the application reminds users to take financial responsibility, such as paying bills on time. In addition, the software tracks personal spending habits and shows results in graphs, allowing projections into future financial scenarios. In addition, the software praises users for doing necessary but menial tasks, like balancing their online check registry. These microsuasion elements—reminders, visualizations, and praise—are influence elements embedded in the Quicken experience in order to change what users think and how they act. Ideally, when these microsuasion elements succeed, users benefit from Quicken's approach to managing personal finances.

Like Quicken, educational software often uses microsuasion. The overall point of most educational applications and interactive experiences is to teach facts and skills, not to persuade. However, in order to get users to stick with the program or to believe what is presented, many products will incorporate motivational elements as well as building credibility perceptions of the content. The product may seek to persuade the learner that the content is important, that the learner is able to successfully master it, and that following the guidelines of the program will lead to the greatest success. Note how these smaller elements of the program—the microsuasions—contribute to the overall goal: learning. Furthermore, interactive educational products will often incorporate elements of games, which leads to a large area related to microsuasion: computer-based gaming.

Video games are typically rich in microsuasion elements. The overall goal of most games is to provide entertainment, not to persuade. However, during the entertainment experience players can be bombarded with microsuasion elements, sometimes continuously. Video games can leverage the seven basic intrinsic motivators: challenge, curiosity, fantasy, control, competition, cooperation, and recognition (Maline & Lepper, 1987). Video games can also incorporate other categories of microsuasion, such as social-influence dynamics.

Captology is relevant to computing products designed with macrosuasion in mind—like Baby Think It Over—and to those that simply use microsuasion in order to make the product more successful—like Quicken. In both cases, designers must understand how to create interactive experiences that change the way people think and behave, whether it is for a single overall outcome or for near-term outcomes that are the building blocks of a larger experience.

NO UNIVERSAL THEORY OF PERSUASION

Creating interactive technology experiences that motivate and influence users would be easy if persuasion were fully understood. It's not. Our understanding is limited, despite the fact that the study of persuasion extends back at least 2,000 years. The fields of psychology, marketing, advertising, public-information campaigns, and others have developed theories and perspectives on how to influence and motivate people, but all approaches have limitations. The reality is this: we have no universal theory or framework for persuasion. In other words, no single set of principles fully explains what motivates people, what causes them to adopt certain attitudes, and what leads them to perform certain behaviors (Fogg, 2003; Ford, 1992). In some ways, this is not a surprise. Human psychology is complex, and persuasion is a large domain, often with fuzzy boundaries. Without a universal theory of persuasion, we must draw from a set of theories and models that describe influence, motivation, or behavior change in specific situations and for specific types of people. This limitation creates an additional challenge for designers of persuasive technology products.

Because computing technology creates new possibilities for influencing people, work in captology can lead to new frameworks, which, although not perfect, enhance the knowledge and practice in HCI. One such framework is the "Functional Triad" (Fogg, 2003).

THE FUNCTIONAL TRIAD: A FRAMEWORK FOR PERSUASIVE TECHNOLOGY

Computers play many roles, some of which go unseen and unnoticed. From a user's perspective, computers function in three basic ways: as (a) tools, as (b) media, and as (c) social actors. In the last two decades, researchers and designers have discussed variants of these functions, usually as metaphors for computer use (i.e., Kay, 1984; Verplank, Fulton, Black, & Moggridge, 1993). However, these three categories are more than metaphors; they are basic ways that people view or respond to computing technologies. These categories also represent three basic types of experiences that motivate and influence people.

Described in more detail elsewhere (Fogg, 1999, 2000, 2003), the Functional Triad is a framework that makes explicit these three computer functions—tools, media, and social actors. First, as this framework suggests, computer applications or systems function as tools, providing users with new abilities or powers. Using computers as tools, people can do things they could not do before, or they can do things more easily.

The Functional Triad also suggests that computers function as media, a role that has grown dramatically during the 1990s as computers became increasingly powerful in displaying graphics and in exchanging information over a network such as the Internet. As a medium, a computer can convey either symbolic content (i.e., text, data graphs, icons) or sensory content (i.e., real-time video, virtual worlds, simulation).

Finally, computers also function as social actors. Empirical research demonstrates that people form social relationships with technologies (Reeves & Nass, 1996). The precise causal factors for these social responses have yet to be outlined in detail, but I propose that users respond socially when computers do at least one of the following: (1) adopt animate characteristics (i.e., physical features, emotions, voice communication), (2) play animate roles (i.e., coach, pet, assistant, opponent), or (3) follow social rules or dynamics (i.e., greetings, apologies, taking turns) (Fogg, 2003).

The Functional Triad is not a theory; it is a framework for analysis and design. In all but the most extreme cases, a single interactive technology is a mix of these three functions, combining them to create an overall user experience.

In captology the Functional Triad is useful because it helps show how computer technologies can employ different techniques for changing attitudes and behaviors. For example, computers as tools persuade differently than computers as social actors. The strategies and theories that apply to each function differ. The paragraphs that follow use the Functional Triad to highlight aspects of persuasive technology, including general design strategies and approaches for creating computing products that persuade and motivate.

Computers as Persuasive Tools

In general, computers as persuasive tools affect attitude and behavior changes by increasing a person's abilities or making something easier to do (Tombari, Fitzpatrick, & Childress, 1985). Although one could propose numerous possibilities for persuasion in this manner, below are four general ways in which computers persuade as tools: by (a) increasing self-efficacy, (b) providing tailored information, (c) triggering decision making, and (d) simplifying or guiding people through a process.

Computers That Increase Self-Efficacy

Computers can increase self-efficacy (Lieberman, 1992), an important contributor to attitude and behavior change processes. Self-efficacy describes individuals' beliefs in their ability to take successful action in specific domains (Bandura, 1997; Bandura, Georgas, & Manthouli, 1996). When people perceive high self-efficacy in a given domain, they are more likely to take action. In addition, because self-efficacy is a perceived quality, even if individuals merely believe that their actions are more effective and productive (perhaps because they are using a specific computing technology), they are more likely to perform a particular behavior (Bandura, 1997; Bandura, Georgas, & Manthousli, 1996). As a result, functioning as tools, computing technologies can make individuals feel more efficient, productive, in

control, and generally more effective (DeCharms, 1968; Kernal, 1999; Pancer, George, & Gebotys, 1992). For example, a heart-rate monitor may help people feel more effective in meeting their exercise goals when it provides ongoing information on heart rate and calories burned. Without the heart-rate monitor, people could still take their pulse and calculate calories, but the computer device—whether it be worn or part of the exercise machinery—makes these tasks easier. The ease of tracking heart rate and calories burned likely increases self-efficacy in fitness behavior, making it more likely the individual will continue to exercise (Brehm, 1997; Strecher, DeVellis, Becker, & Rosenstock, 1986; Thompson, 1992).

Computers That Provide Tailored Information

Next, computers act as tools when they tailor information, offering people content that is pertinent to their needs and contexts. Compared to general information, tailored information increases the potential for attitude and behavior change (Beniger, 1987; Dijkstra, Librand, & Timminga., 1998; Jimison, Street, & Gold, 1997; Nowak, Shamp, Hollander, Cameron, Schumann, & Thorson, 1999; Strecher, 1999; Strecher, Kreuter, Den Boer, Kobrin, Hospers, & Skinner., 1994).

One notable example of a tailoring technology is the Website discussed earlier, Chemical Scorecard (www.scorecard.org), which generates information according to an individual's geographical location in order to achieve a persuasive outcome. After people enter their zip code into this Website, the Web technology reports on chemical hazards in their neighborhood, identifies companies that create those hazards, and describes the potential health risks. Although no published studies document the persuasive effects of this particular technology, outside research and analysis suggests that making information relevant to individuals increases their attention and arousal, which can ultimately lead to increased attitude and behavior change (Beniger, 1987; MacInnis & Jaworski, 1989; MacInnis, Moorman, & Jaworski, 1991; Strecher, 1999).

Computers That Trigger Decision-Making

Technology can also influence people by triggering or cueing a decision-making process. For example, today's web browsers launch a new window to alert people before they send information over insecure network connections. The message window serves as a signal to consumers to rethink their planned actions. A similar example exists in a very different context. Cities concerned with automobile speeding in neighborhoods can use a stand-alone radar trailer that senses the velocity of an oncoming automobile and displays that speed on a large screen. This technology is designed to trigger a decision-making process regarding driving speed.

Computers That Simplify or Guide People Through a Process

By facilitating or simplifying a process for users, technology can minimize barriers that may impede a target behavior. For example, in the context of web commerce, technology can simplify a multistep process down to a few mouse clicks. Typically, in order to purchase something online, a consumer needs to select an item, place it in a virtual shopping cart, proceed to checkout, enter personal and billing information, and verify an order confirmation. Amazon.com and other e-commerce companies have simplified this process by storing customer information so that consumers need not reenter information every transaction. By lowering the time commitment and reducing the steps to accomplish a goal, these companies have reduced the barriers for purchasing products from their sites. The principle used by Web and other computer technology (Todd & Benbasat, 1994) is similar to the dynamic Ross and Nisbett (1991) discussed on facilitating behaviors through modifying the situation.

In addition to reducing barriers for a target behavior, computers can also lead people through processes to help them change attitudes and behaviors (Muehlenhard, Baldwin, Bourg, & Piper, 1988; Tombari, Fitzpatrick, & Childress, 1985). For example, a computer nutritionist can guide individuals through a month of healthy eating by providing recipes for each day and grocery lists for each week. In general, by following a computer-led process, users (a) are exposed to information they may not have seen otherwise, and (b) are engaged in activities they may not have done otherwise (Fogg, 2000, 2003).

Computers as Persuasive Media

The next area of the Functional Triad deals with computers as persuasive media. Although "media" can mean many things, here the focus is on the power of computer simulations. In this role computer technology provides people with experiences, either first-hand or vicarious. By providing simulated experiences, computers can change people's attitudes and behaviors. Outside the world of computing, experiences have a powerful impact on people's attitudes, behaviors, and thoughts (Reed, 1996). Experiences offered via interactive technology have similar effects (Bullinger, Roessler, Mueller-Spahn, Riva, & Wiederhold, 1998; Fogg, 2000).

Three types of computer simulations are relevant to persuasive technologies:

- simulated cause-and-effect scenarios
- simulated environments
- simulated objects

The paragraphs that follow discuss each simulation type in turn. (Note that other taxonomies for simulations exist. For example, see Gredler (1986), de Jong (1991), and Alessi (1991)).

Computers That Simulate Cause and Effect

One type of computer simulation allows users to vary the inputs and observe the effects (Hennessy & O'Shea, 1993)—what one could call "cause-and-effect simulators." The key to effective cause-and-effect simulators is their ability to demonstrate the consequence of actions immediately and credibly (Alessi, 1991;

Balci, 1998; Balci, Henrikson, & Roberts, 1986; Crosbie & Hay, 1978; de Jong, 1991; Hennessy & O'Shea, 1993; Zietsman & Hewson, 1986). These computer simulations give people first-hand insight into how inputs (such as putting money in a savings account) affect an output (such as accrued retirement savings). By allowing people to explore causes and effects of situations, these computer simulations can shape attitudes and behaviors.

Computers That Simulate Environments

A second type of computer simulation is the environment simulator. These simulators are designed to provide users with new surroundings, usually through images and sound. In these simulated environments, users have experiences that can lead to attitude and behavior change (Bullinger et al., 1998), including experiences that are designed as games or explorations (Lieberman, 1992; Schlosser & Kanifer, 1999; Schneider, 1985; Woodward, Carnine, & Davis, 1986).

The efficacy of this approach is demonstrated by research on the Tectrix Virtual Reality Bike (an exercise bike that includes a computer and monitor that shows a simulated world). Porcari and colleagues (1998) found that people using an exercise device with computer simulation of a passing landscape exercised harder than those who used an exercise device without simulation. Both groups, however, felt that they had exerted themselves a similar amount. This outcome caused by simulating an outdoor experience mirrors findings from other research: people exercise harder when outside than inside a gym (Ceci & Hassmen, 1991).

Environmental simulators can also change attitudes. Using a virtual reality environment in which the people saw and felt a simulated spider, Carlin and colleagues (1997) were able to decrease the fear of spiders in his participants. In this research, participants wore a head-mounted display that immersed them into a virtual room, and they were able to control both the number of spiders and their proximity. In this case study, Carlin found that the virtual reality treatment reduced the fear of spiders in the real world. Other similar therapies have been used for fear of flying (Klein, 1999; Wiederhold, Davis, Wiederhold, & Riva, 1998), agoraphobia (Ghosh & Marks, 1987), claustrophobia (Bullinger et al., 1998), and fear of heights (Bullinger), among others (Kirby, 1996).

Computers That Simulate Objects

The third type of computer simulations are "object simulators." These computerized devices simulate an object (as opposed to an environment). The Baby Think It Over infant simulator described earlier in this chapter is one such device. Another example is a specially equipped car created by Chrysler Corporation, designed to help teens experience the effect of alcohol on their driving. Used as part of high-school programs, teen drivers first navigate the special car under normal conditions. Then the operator activates an onboard computer system, which simulates how an inebriated person would drive—breaking sluggishly, steering inaccurately, and so on. This computer-enhanced care provides teens with an experience designed to change their attitudes and behaviors about drinking and driving. Although the sponsors of this car do not measure the impact of this intervention, the anecdotal evidence is compelling (i.e., see Machrone, 1998).

Table 7.2 lists the three types of simulations just discussed above and outlines what advantage each type of simulation offers as far as persuasion and motivation are concerned.

Computers as Persuasive Social Actors

The final corner of the Functional Triad focuses on computers as "persuasive social actors," a view of computers that has only recently become widely recognized. Past empirical research has shown that individuals form social relationships with technology, even when the stimulus is rather impoverished (Fogg, 1997; Marshall & Maguire, 1971; Moon & Nass, 1996; Muller, 1974; Nass, Fogg, & Youngme, 1996; Nass, Moon, Fogg, Reeves, & Dryer, 1995; Nass & Steuer, 1993; Nass, Youngme, Morkes, Eun-Young, & Fogg, 1997; Parise, Kiesler, Sproull, & Waters, 1999; Quintanar Crowell, & Pryor 1982; Reeves & Nass, 1996). For example, individuals share reciprocal relationships with computers (Fogg & Nass, 1997a; Parise, Keisler, Sproull, & Waters, 1999), can be flattered by computers (Fogg & Nass, 1997b), and are polite to computers (Nass, Moon, & Carney, 1999).

In general we propose that, computers as social actors can persuade people to change their attitudes and behaviors by (a) providing social support, (b) modeling attitudes or behaviors, and (c) leveraging social rules and dynamics (Fogg, 2003).

Computers That Provide Social Support

Computers can provide a form of social support in order to persuade, a dynamic that has long been observed in human-

TABLE 7.2. Captology Includes Three Types of Persuasive Simulations

Simulation Type	Key Advantages
Cause-and-effect simulators	• Allow users to explore and experiment • Show cause-and-effect relationships clearly and quickly • Persuade without being overly didactic
Environment simulators	• Can create situations that reward and motivate people for a target behavior • Allow rehearsal: practicing a target behavior • Can control exposure to new or frightening situations • Facilitate role playing: adopting another person's perspective
Object simulators	• Fit into the context of a person's normal life • Are less dependent on imagination or suspension of disbelief • Make clear the impact on normal life

human interactions (Jones, 1990). While the potential for effective social support from computer technology has yet to be fully explored, a small set of empirical studies provide evidence for this phenomenon (Fogg, 1997; Fogg & Nass, 1997b; Nass et al., 1996; Reeves & Nass, 1996). For example, computing technology can influence individuals by providing praise or criticism, thus manipulating levels of social support (Fogg & Nass, 1997b; Muehlenhard et al., 1988).

Outside the research context, various technology products use the power of praise to influence users. For example, the Dole *5 A Day* CD-ROM, discussed earlier, uses a cast of over 30 onscreen characters to provide social support to users who perform various activities. Characters such as "Bobby Banana" and "Pamela Pineapple" praise individuals for checking labels on virtual frozen foods, for following guidelines from the food pyramid, and for creating a nutritious virtual salad.

Computers That Model Attitudes and Behaviors

In addition to providing social support, computer systems can persuade by modeling target attitudes and behaviors. In the natural world, people learn directly through first-hand experience and indirectly through observation (Bandura, 1997). When a behavior is modeled by an attractive individual or is shown to result in positive consequences, people are more likely to enact that behavior (Bandura). Lieberman's research (1997) on a computer game designed to model health-maintenance behaviors shows the positive effects that an onscreen cartoon model had on those who played the game. In a similar way, the product "Alcohol 101" (www.centurycouncil.org/underage/education/a101.cfm) uses navigable onscreen video clips of human actors dealing with problematic situations that arise during college drinking parties. The initial studies on the Alcohol 101 intervention show positive outcomes (Reis, 1998). Computer-based characters, whether artistically rendered or video images, are increasingly likely to serve as models for attitudes and behaviors.

Computers That Leverage Social Rules and Dynamics

Computers have also been shown to be effective persuasive social actors when they leverage social rules and dynamics (Fogg, 1997; Friedman & Grudin, 1998; Marshall & Maguire, 1971; Parise et al., 1999). These rules include turn taking, politeness norms, and sources of praise (Reeves & Nass, 1996). The rule of reciprocity— that we must return favors to others— is among the most powerful social rule (Gouldner, 1960) and is one that has also been shown to have force when people interact with computers. Fogg and Nass (1997a) showed that people performed more work and better work for a computer that assisted them on a previous task. In essence, users reciprocated help to a computer. On the retaliation side, the inverse of reciprocity, the research showed that people performed lower quality work for a computer that had served them poorly in a previous task. In a related vein, Moon (1998) found that individuals followed rules of impression management when interacting with a computer. Specifically, when individuals believed that the computer interviewing them was in the same room, they provided more honest answers, compared to interacting with a computer believed to be a few miles away. In addition, subjects were more persuaded by the proximate computer.

The previous paragraphs outline some of the early demonstrations of computers as social actors that motivate and influence people in predetermined ways, often paralleling research from long-standing human-human research.

Functional Triad Summary

Table 7.3 summarizes the Functional Triad and the persuasive affordances that each element offers.

In summary, the Functional Triad can be a useful framework in captology, the study of computers as persuasive technologies. It makes explicit how a technology can change attitudes and behaviors—either by increasing a person's capability, by providing users with an experience, or by leveraging the power of social relationships. Each of these paths suggests related persuasion strategies, dynamics, and theories. One element that is common to all three functions is the role of credibility. Credible tools, credible media, and credible social actors will all lead to increased power to persuade. This is the focus of the next section.

COMPUTERS AND CREDIBILITY

One key issue in captology is computer credibility, a topic that suggests questions such as, "Do people find computers to be credible sources?," "What aspects of computers boost credibil-

TABLE 7.3. Captology Outlines Three Ways That Computers Influence People

Function	Essence	Persuasive Affordances
Computer as **tool** or **instrument**	Increases capabilities	• Reduces barriers (time, effort, cost) • Increases self-efficacy • Provides information for better decision making • Changes mental models
Computer as **medium**	Provides experiences	• Provides first-hand learning, insight, visualization, resolve • Promotes understanding of cause-and-effect relationships • Motivates through experience, sensation
Computer as **social actor**	Creates relationship	• Establishes social norms • Invokes social rules and dynamics • Provides social support or sanction

ity?," and "How do computers gain and lose credibility?" Understanding the elements of computer credibility promotes a deeper understanding of how computers can change attitudes and behaviors, as credibility is a key element in many persuasion processes (Gahm, 1986; Lerch & Prietula, 1989; Lerch, Prietula, & Kulik, 1997).

Credibility has been a topic of social science research since the 1930s (for reviews, see Petty & Cacioppo, 1981; Self, 1996). Virtually all credibility researchers have described credibility as a perceived quality made up of multiple dimensions (i.e., Buller & Burgoon, 1996; Gatignon & Robertson, 1991; Petty & Cacioppo, 1981; Self, 1996; Stiff, 1994). This description has two key components germane to computer credibility. First, credibility is a perceived quality; it does not reside in an object, a person, or a piece of information. Therefore, in discussing the credibility of a computer product, one is always discussing the *perception* of credibility for the computer product.

Next, researchers generally agree that credibility perceptions result from evaluating multiple dimensions simultaneously. Although the literature varies on exactly how many dimensions contribute to the credibility construct, the majority of researchers identify trustworthiness and expertise as the two key components of credibility (Self, 1996). Trustworthiness, a key element in the credibility calculus, is described by the terms *well intentioned*, *truthful*, *unbiased*, and so on. The trustworthiness dimension of credibility captures the perceived goodness or morality of the source. Expertise, the other dimension of credibility, is described by terms such as *knowledgeable*, *experienced*, *competent*, and so on. The expertise dimension of credibility captures the perceived knowledge and skill of the source.

Extending research on credibility to the domain of computers, we have proposed that *highly credible computer products will be perceived to have high levels of both trustworthiness and expertise* (Fogg & Tseng, 1999). In evaluating credibility, a computer user will assess the computer product's trustworthiness and expertise to arrive at an overall credibility assessment.

When Does Credibility Matter?

Credibility is a key component in bringing about attitude change. Just as credible people can influence other people, credible computing products also have the power to persuade. Computer credibility is not an issue when there is no awareness of the computer itself or when the dimensions of computer credibility—trustworthiness and expertise—are not at stake. In these cases computer credibility does not matter to the user. However, in many cases credibility is key. The following seven categories outline when credibility matters in HCI (Tseng & Fogg, 1999).

1. When computers act as a knowledge repository. Credibility matters when computers provide data or knowledge to users. The information can be static information, such as simple web pages or an encyclopedia on CD-ROM. But computer information can also be dynamic. Computers can tailor information in real time for users, such as providing information that matches interests, personality, or goals. In such cases, users may question the credibility of the information provided.

2. When computers instruct or tutor users. Computer credibility also matters when computers give advice or provide instructions to users. Sometimes it's obvious why computers give advice. For example, auto-navigation systems give advice about which route to take, and online help systems advise users on how to solve a problem. These are clear instances of computers giving advice. However, at times the advice from a computing system is subtle. For example, interface layout and menu options can be a form of advice. Consider a default button on a dialogue box. The fact that one option is automatically selected as the default option suggests that certain paths are more likely or profitable for most users. One can imagine that if the default options are poorly chosen, the computer program could lose some credibility.

3. When computers report measurements. Computer credibility is also at stake when computing devices act as measuring instruments. These can include engineering measurements (i.e., an oscilloscope), medical measurements (i.e., a glucose monitor), geographical measurements (i.e., devices with GPS technology), and others. In this area we observed an interesting phenomenon in the 1990s when digital test and measurement equipment was created to replace traditional analog devices. Many engineers, usually those with senior status, did not trust the information from the digital devices. As a result, some engineers rejected the convenience and power of the new technology because their old analog equipment gave information they found more credible.

4. When computers report on work performed. Computers also need credibility when they report to users on work the computer has performed. For example, computers report the success of a software installation or the eradication of viruses. In these cases and others, the credibility of the computer is at issue if the work the computer reports do not match what actually happened. For example, suppose a user runs a spell check and the computer reports no misspelled words. If the user later finds a misspelled word, then the credibility of the program will suffer.

5. When computers report about their own state. Computers also report their own state, and these reports have credibility implications. For example, computers may report how much disk space they have left, how long their batteries will last, how long a process will take, and so on. A computer reporting about its own state raises issues about its competence in conveying accurate information about itself, which is likely to affect user perceptions of credibility.

6. When computers run simulations. Credibility is also important when computers run simulations. This includes simulations of aircraft navigation, chemical processes, social dynamics, nuclear disasters, and so on. Simulations can show cause-and-effect relationships, such as the progress of a disease in a population or the effects of global warming. Similarly, simulations can replicate the dynamics of an experience, such as piloting an aircraft or caring for a baby. Based on rules that humans provide, computer simulations can be flawed or biased. Even if the bias is not intentional, when users perceive that the

computer simulation lacks veridicality, the computer application will lose credibility.

7. When computers render virtual environments. Related to simulations is the computer's ability to create virtual environments for users. Credibility is important in making these environments believable, useful, and engaging. However, virtual environments don't always need to match the physical world; they simply need to model what they propose to model. For example, like good fiction or art, a virtual world for a fanciful arcade game can be highly credible if the world is internally consistent.

Web Credibility Research and Guidelines for Design

When it comes to credibility, the Web is unusual. The Web can be the most credible source of information, and the Web can be among the least credible sources. Limitations inherent to traditional media—most notably modality, interactivity, and space limitations—are often avoidable on the Web. As a result, online information has the potential to be more complete and enhanced by interactive options for users to more thoughtfully process what they read.

However, this potential is accompanied by several features of the Web that can erode its credibility as a medium (Danielson, 2005). First, the Web lacks the traditional gate keeping and quality-control mechanisms that are commonplace to more traditional publishing, such as editing and fact checking. Second, because digital information can be manipulated, disseminated, and published with relative ease, online information seekers must learn to account for incorrect information being widely and quickly duplicated (Metzger, Flanagin, & Zwarun, 2003), as in the case of ubiquitous "Internet hoaxes." Third, where in most media environments prior to the Web and in face-to-face interactions the speaker or writer of proposed ideas and facts was typically clear to the listener or reader, source *ambiguity* is often the rule rather than the exception in web information seeking. Finally, as with any new media technology, the Web requires users to develop new skills when evaluating various claims (Greer, 2003), as in the case of checking Uniform Resource Locators (URLs) as an indicator of site credibility.

Many Websites today offer users low-quality—or outright misleading— information. As a result, credibility has become a major concern for those seeking or posting information on the Web (Burbules, 2001; Caruso, 1999; Johnson & Kaye, 1998; Kilgore, 1998; McDonald, Schumann, & Thorson, 1999; Nielsen, 1997; Sullivan, 1999). Web users are becoming more skeptical of what they find online and may be wary of Web-based experiences in general.

There's a direct connection between web credibility and persuasion via the Web. When a site gains credibility, it also gains the power to change attitudes, and, at times, behaviors. When a Website lacks credibility, it will not be effective in persuading or motivating users. In few arenas is this connection more direct than in e-commerce, where various online claims and promises about products and services provide the primary or sole basis for buying decisions.

As part of the Persuasive Technology Lab, we have been investigating factors influencing Website credibility and user strategies for making such assessments. A general framework for research in this relatively young field is captured in Fogg's Prominence-Interpretation Theory (Fogg, 2002, 2003). Credibility assessment is an iterative process driven by (a) the likelihood that particular Website elements (such as its privacy policy, advertisements, attractiveness, etc.) will be noticed by an information seeker (*prominence*), and (b) the value that element will be assigned by the user in making a credibility judgment (i.e., increases or decreases perceived credibility) (*interpretation*). Several factors can influence the likelihood of an element being noticed, including the user's level of involvement, the significance of the information sought, and the user's level of Web experience, domain expertise, and other individual differences. Similarly, interpretation is influenced by such individual and contextual factors. Noticeable Website elements are evaluated until either the user is satisfied with an overall credibility assessment, or a constraint (often associated with lack of time or motivation) is reached.

Perhaps more than with any other medium, Web-interaction designers face increasing challenges to design Web experiences that first and foremost hold the attention and motivation of information seekers; the second hill to climb is in persuading Web users to adopt specific behaviors, such as the following:

- register personal information
- purchase things online
- fill out a survey
- click on the ads
- set up a virtual community
- download software
- bookmark the site and return often

If web designers can influence people to perform these actions, they have been successful. These are key behavioral outcomes. But what do users notice when evaluating web content, and how are those noticed elements interpreted? What makes a Website credible? We offer the following broad guidelines, arising out of our lab's experimental work:

Guideline #1: Design websites to convey the "real world" aspect of the organization. Perhaps the most effective way to enhance the credibility of a Website is to include elements that highlight the brick-and-mortar nature of the organization it represents. Despite rampant source ambiguity on the Web, web users show a strong reliance on indicators of identity (Rieh, 2002), including credentials, photos, and contact information (Fogg, Marshall, Laraki, Osipovich, Varma, Fang, et al., 2001). The overall implication seems clear: To create a site with maximum credibility, designers should highlight features that communicate the legitimacy and accessibility of the organization.

Guideline #2: Invest resources in visual design. Web users depend to a surprisingly large degree on the visual design of Websites when making credibility judgments. In one study, we found "design look" to be the single most mentioned category by a sample of more than 2,800 users when evaluating the credibility of sites across a wide variety of domains (Fogg,

Soohoo, Danielson, Marable, Stanford, & Tauber, 2003). Similar to the assessment of human communicators, the attractiveness and professional design of a Website is often used as a first indicator of credibility.

Guideline #3: Make Websites easy to use. In the HCI community we have long emphasized ease of use, so a guideline advocating ease of use is not new. However, our work adds another important reason for making Websites usable: it will enhance the site's credibility. In one study (Fogg et al., 2001), people awarded a Website credibility points for being usable (i.e., "The site is arranged in a way that makes sense to you"), and they deducted credibility points for ease-of-use problems (i.e., "the site is difficult to navigate"). While this information should not change how we, as HCI professionals, design user experiences for the Web, it does add a compelling new reason for investing time and money in usable design—it makes a site more credible. Going beyond the data, one could reasonably conclude that a simple, usable Website would be perceived as more credible than a site that has extravagant features but is lacking in usability.

Guideline #4: Include markers of expertise. Expertise is a key component in credibility, and our work supports the idea that Websites that convey expertise can gain credibility in users' eyes. Important "expertise elements" include listing an author's credentials and including citations and references. It's likely that many other elements also exist. Many Websites today miss opportunities to convey legitimately expertise to their users.

Guideline #5: Include markers of trustworthiness. Trustworthiness is another key component in credibility. As with expertise, Website elements that convey trustworthiness will lead to increased perceptions of credibility. Such elements include linking to outside materials and sources, stating a policy on content, and so on. Making information verifiable on a Website increases credibility despite the fact that users are unlikely to follow through on verification (Metzger et al., 2003). Thus, the mere presence of some design elements will influence user perceptions. We propose that Website designers who concentrate on conveying the honest, unbiased nature of their Website will end up with a more credible—and therefore more effective—Website.

Guideline #6: Tailor the user experience. Although not as vital as the previous suggestions, tailoring does make a difference. Our work shows that tailoring the user experience on a Website leads to increased perceptions of web credibility. For example, people think a site is more credible when it acknowledges that the individual has visited it before. To be sure, tailoring and personalization can take place in many ways. Tailoring extends even to the type of ads shown on the page: ads that match what the user is seeking seem to increase the perception of Website credibility.

Guideline #7. Avoid overly commercial elements on a Website. Although most Websites, especially large Websites, exist for commercial purposes, our work suggests that users penalize sites that have an aggressively commercial flavor. For example, web pages that mix ads with content to the point of confusing readers will be perceived as not credible. Fogg et al. (2001) found that mixing ads and content received the most negative response of all. However, it is important to note that ads don't always reduce credibility. In this study and elsewhere (Kim, 1999), quantitative research shows that banner ads done well can enhance the perceived credibility of a site. It seems reasonable that, as with other elements of people's lives, we accept commercialization to an extent but become wary when it is overdone.

Guideline #8. Avoid the pitfalls of amateurism. Most web designers seek a professional outcome in their work. Organizations that care about credibility should be ever vigilant—and perhaps obsessive—to avoid small glitches in their Websites. These "small" glitches seem to have a large impact on web credibility perceptions. Even one typographical error or a single broken link is damaging. While designers may face pressures to create dazzling technical features on Websites, failing to correct small errors undermines that work.

Despite the growing body of research, much remains to be discovered about web credibility. The study of web credibility needs to be an ongoing concern because three things continue to evolve: (a) Web technology, (b) the type of people using the Web, and (c) people's experiences with the Web. Fortunately, what researchers learn about designing for web credibility can translate into credible experiences in other high-tech devices that share information, from mobile phones to gas pumps.

POSITIVE AND NEGATIVE APPLICATIONS OF PERSUASIVE TECHNOLOGY

As the power of persuasive techniques becomes more understood, we are beginning to see more examples of persuasive technologies being created. Many have positive goals in mind, but there are also many technologies designed to negatively influence attitudes and behaviors.

An example of a positive technology is a mobile application called MyFoodPhone. Mobile persuasive devices have several unique properties that may improve their abilities to persuade. First, they are personal devices: people carry their mobile phones everywhere, customize them, and store personal information in them. Second, intrinsic to them being mobile, these devices have the potential to intervene at the right moment, a concept called *kairos*.

MyFoodPhone is an application for the camera phone that helps people watch what they eat—whether they want to change their weight or just eat right. Whenever a user is concerned with an item they are about to eat, they simply take a picture of it with their camera, then use MyFoodPhone to send it to a system that shares the images with a expert dietician. The user receives expert evaluation and feedback. In this case, the simplicity and appropriate timing of the application make it a powerful persuasive tool.

On the Web, GoDaddy (www.godaddy.com), a popular Web-hosting company, attempts to persuade users to purchase more expensive hosting solutions by "disguising" links to their less-expensive plans with plain text links, while links to more pricey upgrades are in large, brightly colored buttons.

A more negative example can be found in the rise in "Pro Anorexia" Websites, encouraging self-starvation and sharing tips for losing weight. Though they reached their height in earlier part of the decade, many of these sites are still being operated. By creating social networks around it, people suffering from anorexia are supported and encouraged to continue their unhealthy habits. Many of these Websites use the web credibility techniques discussed earlier: the sites are well designed and contain expert advice.

As the power of persuasive technologies becomes more understood, the consideration of the ethical ramifications of these technologies becomes essential.

THE ETHICS OF COMPUTING SYSTEMS DESIGNED TO PERSUADE

In addition to research and design issues, captology addresses the ethical issues that arise from design or distributing persuasive interactive technologies. Persuasion is a value-laden activity. By extension, creating or distributing an interactive technology that attempts to persuade is also value laden. Ethical problems arise when the values, goals, and interests of the creators don't match with those of the people who use the technology. HCI professionals can ask a few key questions to get insight into possible ethical problem areas:

- Does the persuasive technology advocate what's good and fair?
- Is the technology inclusive, allowing access to all, regardless of social standing?
- Does it promote self-determination?
- Does it represent what's thought to be true and accurate?

Answering no to any of these questions suggests the persuasive technology at hand could be ethically questionable and perhaps downright objectionable (for a longer discussion on ethics, see Friedman & Kahn, later in this volume).

While it's clear that deception and coercion are unethical in computing products, some behavior change strategies such as conditioning, surveillance, and punishment, are less cut and dry. For example, Operant conditioning—a system of rewards—can powerfully shape behaviors. By providing rewards, a computer product could get people to perform new behaviors without their clear consent or without them noticing the forces of influence at work.

Surveillance is another common and effective way to change behavior. People who know they are being watched behave differently. Today, computer technologies allow surveillance in ways that were never before possible, giving institutions remarkable new powers. Although advocates of computer-based employee surveillance (i.e., DeTienne, 1993) say that monitoring can "inspire employees to achieve excellence," they and opponents agree that such approaches can hurt morale or create a more stressful workplace. When every keystroke and every restroom break is monitored and recorded, employees may feel they are part of an electronic sweatshop.

Another area of concern is when technologies use punishment—or threats of punishment—to shape behaviors. Although punishment is an effective way to change outward behaviors in the short term, punishment has limited outcomes beyond changing observable behavior, and many behavior change experts frown on using it. The problems with punishment increase when a computer product punishes people. The punishment may be excessive or inappropriate to the situation. Also, the long-term effects of punishment are likely to be negative. In these cases, who bears responsibility for the outcome?

Discussed elsewhere in more detail (Berdichevsky, 1999; Fogg, 1998, 2003), those who create or distribute persuasive technologies have a responsibility to examine the moral issues involved.

PERSUASIVE TECHNOLOGY: POTENTIAL AND RESPONSIBILITY

Computer systems are now becoming a common part of everyday life. Whatever the form of the system, from a desktop computer to a smart car interior to a mobile phone, these interactive experiences can be designed to influence our attitudes and affect our behaviors. They can motivate and persuade by merging the power of computing with the psychology of persuasion.

We humans are still the supreme agents of influence—and this won't change any time soon. Computers are not yet as effective as skilled human persuaders are, but at times computing technology can go beyond what humans can do. Computers never forget, they don't need to sleep, and they can be programmed to never stop trying. For better or worse, computers provide us with a new avenue for changing how people think and act.

To a large extent, we as a community of HCI professionals will help create the next generation of technology products, including those products designed change people's attitudes and behaviors. If we take the right steps—raising awareness of persuasive technology in the general public and encouraging technology creators to follow guidelines for ethical interactive technologies—we may well see persuasive technology reach its potential, enhancing the quality of life for individuals, communities, and society.

References

Alessi, S. M. (1991). Fidelity in the design of Instructional simulations. *Journal of Computer-Based Instruction, 15*, 40–47.

Balci, O. (1998, December 13–16). *Verification, validation, and accreditation*. Paper presented at the Winter Simulation Conference, Washington United States.

Balci, O., Henrikson, J. O., & Roberts, S. D. (1986). Credibility assessment of simulation results. In J. R. Wilson (Ed.), Winter Simulation Conference Proceedings.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W. H. Freeman.

Bandura, A., Georgas, J., & Manthouli, M. (1996). Reflections on human agency, *Contemporary psychology in Europe: Theory, research, and applications.* Seattle, WA: Hogrefe & Huber.

Beniger, J. R. (1987). Personalization of mass media and the growth of pseudo-community. *Communication Research, 14*(3), 352–371.

Berdichevsky, D., & Neunschwander, E. (1999). Towards an ethics of persuasive technology. *Communications of the ACM, 42*(5), 51–8.

Brehm, B. (1997, December). Self-confidence and exercise success. *Fitness Management,* 22–23.

Buller, D. B., & Burgoon, J. K. (1996). Interpersonal Deception Theory. *Communication Theory, 6*(3), 203–242.

Bullinger, A. H., Roessler, A., Mueller-Spahn, F., Riva, G., & Wiederhold, B. K. (1998). From toy to tool: The development of immersive virtual reality environments for psychotherapy of specific phobias, *Studies in Health Technology and Informatics, 58.* Amsterdam: IOS Press.

Burbules, N. C. (2001). Paradoxes of the Web: The ethical dimensions of credibility. *Library Trends, 49*, 441–453.

Carlin, A. S., Hoffman, H. G., & Weghorst, S. (1997). Virtual reality and tactile augmentation in the treatment of spider phobia: A case report. *Behaviour Research & Therapy, 35*(2), 153–158.

Caruso, D. (1999, November 22). Digital commerce: Self indulgence in the Internet industry. *The New York Times*.

Ceci, R., & Hassmen, P. (1991). Self-monitored exercise at three different PE intensities in treadmill vs. field running. *Medicine and Science in Sports and Exercise, 23*, 732–738.

Crosbie, R. E., & Hay, J. L. (1978). The credibility of computerised models. In R. E. Crosbie (Ed.), *Toward real-time simulation: Languages, models and systems*. La Jolla, CA: Society of Computer Simulation.

Danielson, D. R. (2005). Web credibility. To appear in C. Ghaoui (Ed.), *Encyclopedia of human-computer interaction*. Idea Group.

DeCharms, R. (1968). *Personal causation: The internal affective determinants of behavior*. New York: Academic.

de Jong, T. (1991). Learning and instruction with computer simulations. *Education & Computing, 6*, 217–229.

Denning, P., & Metcalfe, R. (1997). *Beyond calculation: The next fifty years of computing*. New York: Springer-Verlag.

DeTienne, Kristen Bell. (1993, September/October). Big Brother or friendly coach? Computer monitoring in the 21st century. *The Futurist*.

Dijkstra, J. J., Liebrand, W. B. G., & Timminga, E. (1998). Persuasiveness of expert systems. *Behaviour and Information Technology, 17*(3), 155–63.

Fogg, B.J. (1997). *Charismatic computers: Creating more likable and persuasive interactive technologies by leveraging principles from social psychology* (Doctoral Thesis. Stanford University).

Fogg, B.J. (1998). Persuasive computers: Perspectives and research directions. *Proceedings of the Conference on Human Factors in Computing Systems*. CHI98, Los Angeles, USA.

Fogg, B.J. (1999). Persuasive technologies. *Communications of the ACM, 42*(5), 26–29.

Fogg, B.J. (2002). *Prominence-interpretation theory: Explaining how people assess credibility*. Research report from the Stanford Persuasive Technology Lab, Stanford University. Retrieved (date), from at: http://credibility.stanford.edu/

Fogg, B.J. (2003a). *Persuasive technology: Using computers to change what we think and do*. San Francisco: Morgan-Kaufmann.

Fogg, B.J. (2003b). Prominence-interpretation theory: Explaining how people assess credibility online. *Proceedings of CHI'03, Extended Abstracts on Human Factors in Computing Systems* (pp. 722–723).

Fogg, B.J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., et al. (2000). Elements that affect web credibility: Early results from a self-report study. *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems.* New York: ACM Press.

Fogg, B.J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., et al. (2001). What makes a Website credible? A report on a large quantitative study. *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems.* New York: ACM Press.

Fogg, B.J., & Nass, C. (1997a). How users reciprocate to computers: An experiment that demonstrates behavior change. *Proceedings of the Conference on Human Factors in Computing Systems,* CHI 97. New York: ACM.

Fogg, B.J., & Nass, C. (1997b). Silicon sycophants: The effects of computers that flatter. *International Journal of Human-Computer Studies, 46*(5), 551–61.

Fogg, B.J., Soohoo, C., Danielson, D. R., Marable, L., Stanford, J., & Tauber, E. R. (2003). How do users evaluate the credibility of Websites? A study with over 2,500 participants. *Proceedings of DUX2003, Designing for User Experiences Conference*.

Fogg, B.J., & Tseng, H. (1999). The elements of computer credibility. *Proceedings of the Conference on Human Factors and Computing Systems*, CHI99. Pittsburgh, PA USA.

Ford, M. E. (1992). *Motivating humans: Goals, emotions, personal agency beliefs*. Newbury Park: Sage.

Friedman, B., & Grudin, J. (1998). Trust and accountability: Preserving human values in interactional experience. *Proceedings of the Conference on Human Factors in Computing Systems*, CHI98, Los Angeles, USA.

Gahm, G. A. (1986). *The effects of computers, source salience and credibility on persuasion.* Doctoral thesis. State University of New York.

Gatignon, H., & Robertson, T. S. (1991). *Innovative Decision Processes*. Englewood Cliffs, NJ: Prentice-Hall.

Ghosh, A., & Marks, I. M. (1987). Self-treatment of agoraphobia by exposure. *Behavior Therapy, 18*(1), 3–16.

Gouldner, A. W. (1960). The Norm of reciprocity: A preliminary statement. *American Sociological Review, 25*, 161–178.

Gredler, M. B. (1986). A taxonomy of computer simulations. *Educational Technology, 26*, 7– 12.

Greer, J. D. (2003). Evaluating the credibility of online information: A test of source and advertising influence. *Mass Communication & Society, 6*(1), 11–28.

Hennessy, S., & O'Shea, T. (1993). Learner perceptions of realism and magic in computer simulations. *British Journal of Educational Technology, 24*(2), 125–38.

Jimison, H. B., Street, R. L., Jr., & Gold, W. R. (1997). Patient-specific interfaces to health and decision-making information, *LEA's communication series*. Mahwah, NJ: Lawrence Erlbaum.

Johnson, T. J., & Kaye, B. K. (1998). Cruising is believing?: Comparing Internet and traditional sources on Media Credibility Measures. *Journalism and Mass Communication Quarterly, 75*(2), 325–40.

Jones, E. E. (1990). *Interpersonal perception*. New York: W. H. Freeman.

Kay, A. (1984). Computer software. *Scientific American, 251*, 53–59.

Kernal, H. K. (1999). *Effects of design characteristics on evaluation of a home control system: A comparison of two research methodologies*. Paper presented to SIGGRAPH Annual Conference 2000, New Orleans, LA, USA.

Khaslavsky, J., & Shedroff, N. (1999). Understanding the seductive experience. *Communications of the ACM, 42*(5), 45–9.

Kilgore, R. (1998). Publishers must set rules to preserve credibility. *Advertising Age, 69*(48), 31.

Kim, N. (1999). *World Wide Web credibility: What effects do advertisements and typos have on the perceived credibility of web page information?* Unpublished honors thesis, Stanford University, California.

King, P., & Tester, J. (1999). The landscape of persuasive technologies. *Communications of the ACM, 42*(5), 31–8.

Kirby, K. C. (1996). Computer-Assisted Treatment of Phobias. *Psychiatric Services, 4*(2), 139–140, 142.

Klein, R. A. (1999). Treating fear of flying with virtual reality exposure therapy. *Innovations in Clinical Practice: A Source Handbook, 17*, 449–465.

Lerch, F. J., & Prietula, M. J. (1989). How do we trust machine advice? Designing and using human-computer interfaces and knowledge based systems. *Proceedings of the Third Annual Conference on Human Computer Interaction*.

Lerch, F. J., Prietula, M. J., & Kulik, C. T. (1997). The Turing Effect: The nature of trust in expert system advice. *Expertise in context: Human and machine.* Cambridge, MA: The MIT Press.

Lieberman, D. (1992). The computer's potential role in health education. *Health Communication, 4*, 211–225.

Lieberman, D. (1997). Interactive video games for health promotion. In W. G. R. Street, & T. Mannin (Eds.), *Health promotion and interactive technology (pp.)*. Mahwah, NJ: Lawrence Earlbaum.

Machrone, B. (1998, July 1). Driving drunk. *PC Magazine*.

MacInnis, D. J., & Jaworski, B. J. (1989). Information processing from advertisements: Toward an integrative framework. *Journal of Marketing, 53*(4), 1–23.

MacInnis, D. J., Moorman, C., & Jaworski, B. J. (1991). Enhancing and measuring consumers' motivation, opportunity, and ability to process brand information from ads. *Journal of Marketing, 55*(4), 32–53.

Malone, T., & Lepper, M. (1987). Making learning fun: A taxonomy of intrinsic motivation for learning. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction*. Hillsdale, N.J.: Lawrence Earlbaum.

Marshall, C., & Maguire, T. O. (1971). The computer as social pressure to produce conformity in a simple perceptual task. *AV Communication Review, 19*(1), 19–28.

McDonald, M., Schumann, D. W., & Thorson, E. (1999). Cyberhate: Extending persuasive techniques of low credibility sources to the World Wide Web. In D. W. Schumann & E. Thorson (Eds.), *Advertising and the World Wide Web.* Mahwah: Lawrence Erlbaum.

Metzger, M. J., Flanagin, A. J., & Zwarun, L. (2003). College student Web use, perceptions of information credibility, and verification behavior. *Computers & Education, 41*, 271–290.

Moon, Y. (1998). The effects of distance in local versus remote human-computer interaction. *Proceedings of the Conference on Human Factors in Computing Systems,* CHI 98, (pp. 103–108). New York: ACM.

Moon, Y., & Nass, C. (1996). How "Real" are computer personalities? Psychological responses to personality types in human-computer interaction. *Communication Research, 23*(6), 651–674.

Muehlenhard, C. L., Baldwin, L. E., Bourg, W. J., & Piper, A. M. (1988). Helping women "break the ice: A computer program to help shy women start and maintain conversations with men. *Journal of Computer-Based Instruction, 15*(1), 7–13.

Muller, R. L. (1974). *Conforming to the computer: Social influence in computer-human interaction*. Doctoral thesis, Syracuse University, New York.

Nass, C., Fogg, B.J., & Youngme, M. (1996). Can computers be teammates? *International Journal of Human-Computer Studies, 45*(6), 669–78.

Nass, C., Moon, Y., & Carney, P. (1999). Are people polite to computers? Responses to computer-based interviewing systems. *Journal of Applied Social Psychology, 29*(5), 1093–1110.

Nass, C., Moon, Y., Fogg, B.J., Reeves, B., & Dryer, D. C. (1995). Can computer personalities be human personalities? *International Journal of Human-Computer Studies, 43*(2), 223–39.

Nass, C., & Steuer, J. (1993). Voices, boxes, and sources of messages: Computers and social actors. *Human Communication Research, 19*(4), 504–527.

Nass, C. I., Youngme, M., Morkes, J., Eun-Young, K., & Fogg, B.J. (1997). Computers are social actors: A review of current research. In B. Friedman (Ed.), *Human values and the design of computer technology.* Stanford, CA: CSLI and Cambridge Press.

Nielsen, J. (1997). *How Users Read on the Web*. Retrieved (date), from www.useit.com/alertbox/9710a.html.

Nowak, G. J., Shamp, S., Hollander, B., Cameron, G. T., Schumann, D. W., & Thorson, E. (1999). Interactive media: A means for more meaningful advertising? *Advertising and consumer psychology*. Mahwah: Lawrence Erlbaum.

Pancer, S. M., George, M., & Gebotys, R. J. (1992). Understanding and predicting attitudes toward computers. *Computers in Human Behavior, 8,* 211–222.

Parise, S., Kiesler, S., Sproull, L., & Waters, K. (1999). Cooperating with life-like interface agents. *Computers in Human Behavior, 15*(2), 123–142.

Petty, R. E., & Cacioppo, J. T. (1981). *Attitudes and persuasion: Classic and contemporary approaches*. Dubuque, IA: W.C. Brown.

Porcari, J. P., Zedaker, M. S., & Maldari, M. S. (1998). Virtual Motivation. *Fitness Management,* 48–51.

Quintanar, L., Crowell, C., & Pryor, J. (1982). Human-computer interaction: A preliminary social psychological analysis. *Behavior Research Methods and Instrumentation, 14*(2), 210–220.

Reed, E. (1996). *The Necessity of Experience*. New Haven, CT: Yale University Press.

Reeves, B., & Nass, C. I. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. New York: Cambridge University Press.

Reis, J. (1998). *Research Results: National Data Anaylsis*. Retrieved (Date), from the Century Council Website: www.centurycouncil.org/alcohol101/dem_nat.cfm

Rieh, S. Y. (2002). Judgment of information quality and cognitive authority in the Web. *Journal of the American Society for Information Science and Technology, 53*, 145–161.

Ross, L., & Nisbett, R. E. (1991). *The person and the situation: Perspectives of social psychology*. New York: McGraw-Hill.

Schlosser, A. E., & Kanifer, A. (1999). Current Advertising on the Internet: The Benefits and Usage of Mixed-Media Advertising Strategies. In D. W. Schumann & E. Thorson (Eds.), *Advertising and the World Wide Web* (pp. 41–62). Mahwah, NJ: Lawrence Erlbaum.

Schneider, S. J. (1985). Computer technology and persuasion: the case of smoking cessation, *Proceedings of COMPINT 85: Computer Aided Technologies*. Washington, DC: IEEE.

Self, C. S. (1996). Credibility. In M. Salwen & D. Stacks (Eds.), *An integrated approach to communication theory and research*. Mahwah, NJ: Erlbaum.

Stiff, J. B. (1994). *Persuasive communication*. New York: The Guilford Press.

Strecher, V. J. (1999). Computer-tailored smoking cessation materials: A review and discussion. Special Issue: Computer-tailored education. *Patient Education & Counseling, 36*(2), 107–117.

Strecher, V. J., DeVellis, B. M., Becker, M. H., & Rosenstock, I. M. (1986). The role of self-efficacy in achieving health behavior change. *Health Education Quarterly, 13*(1), 73–92.

Strecher, V. J., Kreuter, M., Den Boer, D.-J., Kobrin, S., Hospers, H. J., & Skinner, C. S. (1994). The effects of computer-tailored smoking cessation messages in family practice settings. *Journal of Family Practice, 39*(3), 262–270.

Sullivan, C. (1999). Newspapers must retain credibility. *Editor & Publisher, 4*.

Thompson, C. A. (1992). *Exercise adherence and performance: Effects on self-efficacy and outcome expectations.* Doctoral thesis.

Todd, P. A., & Benbasat, I. (1994). The influence of decision aids on choice strategies under conditions of high cognitive load. *IEEE Transactions on Systems, Man, & Cybernetics, 24*(4), 537–547.

Tombari, M. L., Fitzpatrick, S. J., & Childress, W. (1985). Using computers as contingency managers in self-monitoring interventions: A case study. *Computers in Human Behavior, 1*(1), 75–82.

Tseng, S., & Fogg, B.J. (1999). Credibility and computing technology. *Communications of the ACM, 42(*5), 39–44.

Verplank, B., Fulton, J., Black, A., & Moggridge, B. (1993). *Observation and invention: Uses of scenarios in interaction design*. Paper presented at INTERCHI'93.

Wiederhold, B. K., Davis, R., Wiederhold, M. D., & Riva, G. (1998). The effects of immersiveness on physiology, *Studies in health technology and informatics, 58*. Amsterdam: IOS.

Woodward, J. P., Carnine, D., & Davis, L. G. (1986). Health ways: A computer simulation for problem solving in personal health management [Special issue: Technological advances in community health]. *Family & Community Health, 9*(2), 60–63.

Zietsman, A. I., & Hewson, P. W. (1986). Effect of Instruction Using Microcomputer Simulations and Conceptual Change Strategies on Science Learning. *Journal of Research in Science Teaching, 23*(1), 27–39.

·8·

HUMAN-ERROR IDENTIFICATION IN HUMAN-COMPUTER INTERACTION

Neville A. Stanton
Brunel University
Runnymede Campus

Human Error **148**
Human-Error Classification **148**
Predicting Human Error **150**
Systematic Human-Error Reduction and Prediction Approach (SHERPA) 150
Task Analysis For Error Identification (TAFEI) 151
Validation of Human-Error Identification **152**
Applying TAFEI to Interface Design **153**
Applying SHERPA to Safety Critical Systems **154**
Conclusions **155**
References **157**

HUMAN ERROR

We are all familiar with the annoyance of errors we make with everyday devices, such as turning on the heat under on an empty kettle, or making mistakes in the programming sequence of our videocassette recorders. People have a tendency to blame themselves for "human error." However, the use and abuse of the term has led some to question the very notion of "human error" (Wagenaar & Groeneweg, 1988). "Human error" is often invoked in the absence of technological explanations. Chapanis (1999) wrote that back in the 1940s he noted that "pilot error" was really "designer error." This was a challenge to contemporary thinking, and showed that design is all-important in human-error reduction. Chapanis became interested in why pilots often retracted the landing gear instead of the landing flaps after landing the aircraft. He identified the problem as designer error rather than pilot error, as the designer had put two identical toggle switches side by side—one for the landing gear, the other for the flaps. Chapanis proposed that the controls should be separated and coded. The separation and coding of controls is now standard human-factors practice. Half a century after Chapanis's original observations, the idea that one can design error-tolerant devices is beginning to gain credence (Baber & Stanton, 1994). One can argue that human error is not a simple matter of one individual making one mistake, so much as the product of a design which has permitted the existence and continuation of specific activities which could lead to errors (Reason, 1990).

Human error is an emotive topic and psychologists have been investigating its origins and causes since the dawn of the discipline (Reason, 1990). Traditional approaches have attributed errors to individuals. Indeed, so-called "Freudian slips" were considered the unwitting revelation of intention, errors revealing what a person was really thinking but did not wish to disclose. More recently, cognitive psychologists have considered the issues of error classification and explanation (Senders & Moray, 1991). The taxonomic approaches of Norman (1988) and Reason (1990) have fostered the development and formal definition of several categories of human error (e.g., capture errors, description errors, data driven errors, associated activation errors, and loss of activation errors) while the work of Reason (1990) and Wickens (1992) attempted to understand the psychological mechanisms which combine to cause errors (e.g., failure of memory, poor perception, errors of decision making, and problems of motor execution). Reason (1990) in particular has argued that we need to consider the activities of the individual if we are to be able to identify what may go wrong. Rather than viewing errors as unpredictable events, this approach regards them to be wholly predictable occurrences based on an analysis of an individual's activities. Reason's definition proposes that errors are "those occasions in which a planned sequence of mental or physical activities fail to achieve its intended outcome, [and] when these failures cannot be attributed to the intervention of some chance agency." (p. 9)

If errors are no longer to be considered as random occurrences, then it follows that we should be able to identify them and predict their likelihood. The impetus to achieve this has been fueled in the wake of several recent and significant incidents, most notably in the nuclear industry where there now exists several human-error identification (HEI) techniques. The aims of this chapter are to,

1. Consider human-error classifications;
2. Look at systems approaches to human error;
3. Consider how human error can be predicted;
4. Examine the validation evidence; and
5. Look at human error in the context of design.

HUMAN-ERROR CLASSIFICATION

The development of formal human-error classification schemes has assisted in the anticipation and analysis of error. The anticipation of error has come about through the development of formal techniques for predicting error, which is dealt with in the Predicting Human Error section. The analysis of error is assisted by taxonomic systems and the interpretation of underlying psychological mechanisms. Three contemporary systems were presented in the work of Norman (1981), Reason (1990), and Wickens (1992).

Norman's (1981) research focused on the categorization of action slips, in which he presented the analysis of 1,000 incidents. Underpinning his analysis was a psychological theory of schema activation. He argued that action sequences are triggered by knowledge structures (organized as memory units and called "schemas"). Within the mind is a hierarchy of schemas that are invoked (or triggered) if particular conditions are satisfied or certain events occur. The theory seems particularly pertinent as a description of skilled behavior. The classification scheme is presented in Table 8.1.

In Neisser's (1976) seminal work "Cognition and Reality," he contended that human thought is closely coupled with a person's interaction with the world. He argued that knowledge of how the world works (e.g., mental models) leads to the anticipation of certain kinds of information, which in turn directs behavior to seek out certain kinds of information and to provide a ready means of interpretation. During this process, as the environment is sampled, the information garnered serves to update and modify the internal, cognitive schema of the world, which will again direct further search. An illustration of the perceptual cycle is shown in Fig. 8.1.

The perceptual cycle can be used to explain human information processing in control rooms. For example—assuming that an individual has the correct knowledge of a videocassette recorder he is programming—his mental model will enable him to anticipate events (such as the menu items he expects to see), search for confirmatory evidence (look at the panel on the video machine), direct a course of action (select a channel, day of the week, start time, end time, etc.), and continually check that the outcome is as expected (menu item and data field respond as anticipated). If the individual uncovers some data he does not expect (such as a menu item not previously encountered, or the data field not accepting his input), he is required to access a source of a wider knowledge of the world to consider possible explanations that will direct future search activities. The completeness of the model is in the description of process

TABLE 8.1. Taxonomy of Slips with Examples

Taxonomy of Slips	Examples of Error Types
Slips that result from errors in the formation of intention	Mode errors: erroneous classification of the situation
	Description errors: ambiguous or incomplete specification of intention
Slips that result from faulty activation of schemas	Capture errors: similar sequences of action, where stronger sequence takes control
	Data-driven activation errors: external events that cause the activation of schemas
	Association-activation errors: currently active schemas that activate other schemas with which they are associated
	Loss-of-activation errors: schemas that lose activation after they have been activated
Slips that result from faulty triggering of active schemas	Blend errors: combination of components from competing schemas
	Premature activation errors: schemas that are activated too early
	Failure to activate errors: failure of the trigger condition or event to activate the schema

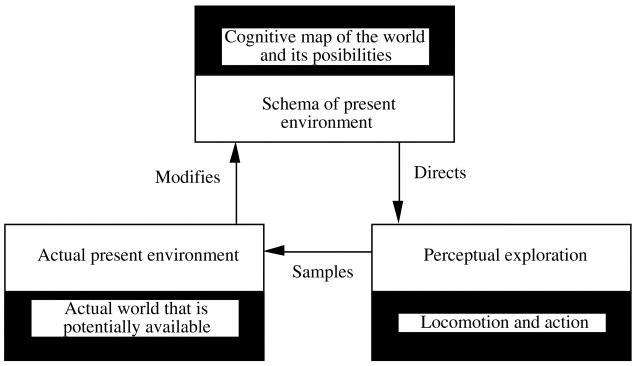


FIGURE 8.1. The perceptual cycle.

(the cyclical nature of sampling the world) and product (the updating of the world model at any point in time).

This interactive schema model works well for explaining how we act in the world. As Norman's (1981) research has shown, it may also explain why errors occur as they do. If, as schema theory predicts, action is directed by schema, then faulty schemas or faulty activation of schemas will lead to erroneous performance. As Table 8.1 shows, this can occur in at least three ways. First, we can select the wrong schema due to misinterpretation of the situation. Second, we can activate the wrong schema because of similarities in the trigger conditions. Third, we can activate schemas too early or too late. Examples of these types of errors are presented in Table 8.1.

Of particular interest is the problem of mode errors. Norman (1981) singled out this type of error as requiring special attention in the design of computing systems. He pointed out that the misclassification of the computing system mode could lead to input errors which may result in serious consequences. In word processors this may mean the loss of documents; in video recorders this may mean the loss of recordings; on flight decks this may mean damage to aircraft.

Casey (1993) described a case in which an apparently simple mode error by a radiotherapy technician working in a cancer care center led to the death of a patient. The Therac-25 she was operating was a state-of-the-art, million-dollar machine that could be used as both a high-power x-ray machine (25 million electron volts delivered by typing an "x" on the keyboard) and low-power electron beam machine (200 rads delivered by typing an "e" on the keyboard). After preparing the patient for radiation therapy, the radiotherapist went to her isolated control room. She accidentally pressed the "x" key instead of the "e" key, but quickly realized this and selected the edit menu so that she could change the setting from x-ray to electron-beam. Then she returned to the main screen to wait for the "beam ready" prompt. All of this occurred within eight seconds. Unknown to her, this rapid sequence of inputs had never been tested on the machine before and it had actually entered a hybrid mode, delivering blasts of 25,000 rads, which was more than 125 times the prescribed dose. When the "beam ready" prompt was displayed, the radiotherapist pressed the "b" key to fire the beam. The high-energy beam was delivered to the patient and the computer screen displayed the prompt "Malfunction 54." Unaware that the machine had already fired, the operator reset it and pressed "b" again. This happened for a third time until the patient ran out of the room reporting painful electric shocks. On investigation, the problem with the machine modes was found, but not before other overdoses had been given. This case study demonstrates the need to consider the way in which design of a system can induce errors in users.

A thorough understanding of human error is required by the design team. Error classification schemes can certainly help, but they need to be supported by formal error-prediction techniques within a user-centered design approach.

Reason (1990) developed a higher-level error classification system, incorporating slips, lapses, and mistakes. Slips and lapses are defined by attentional failures and memory failures, respectively. Both slips and lapses are examples of where the action was unintended, whereas mistakes are associated with intended action. This taxonomy is presented in Table 8.2.

Wickens (1992), taking the information processing framework, considered the implications of psychological mechanisms at work in error formation. He argued that with mistakes, the situation assessment and/or planning are poor while the retrieval action execution is good; with slips, the action execution is poor whereas the situation assessment and planning are good; and finally, with lapses, the situation assessment and ac-

TABLE 8.2. Basic Error Types with Examples

Basic Error Type	Example of Error Type
Slip	Action intrusion
	Omission of action
	Reversal of action
	Misordering of action
	Mistiming of action
Lapse	Omitting of planned actions
	Losing place in action sequence
	Forgetting intended actions
Mistake	Misapplication of good procedure
	Application of a bad procedure
	Misperception
	Poor decision making
	Failure to consider alternatives
	Overconfidence

tion execution are good, but memory is poor. A summary of these distinctions is shown in Table 8.3.

Wickens (1992) was also concerned with mode errors, with particular reference to technological domains. He suggested that a pilot raising the landing gear while the aircraft is still on the runway is an example of a mode error. Wickens proposed that mode errors are the result of poorly conceived system design that allows the mode confusion to occur and the operation in an inappropriate mode. Chapanis (1999) argued that the landing gear switch could be rendered inoperable if the landing gear could be configured to detect weight on the wheels, as the aircraft would be on the ground.

Taxonomies of errors can be used to anticipate what might go wrong in any task. Potentially, every task or activity could be subject to a slip, lapse, or mistake. The two approaches represented within the taxonomies are a schema-based approach and an error-list-based approach. Examples of these two approaches will be presented next in the form of formal human-error identification techniques.

PREDICTING HUMAN ERROR

An abundance of methods for identifying human error exists; some of these methods may be appropriate for the analysis of consumer products. In general, most of the existing techniques have two key problems. The first of these problems relates to the lack of representation of the external environment or objects. Typically, human-error analysis techniques do not represent the activity of the device nor the material that the human interacts with in more than a passing manner. Hollnagel (1993) emphasized that Human Reliability Analysis (HRA) often fails to take adequate account of the context in which performance occurs. Second, there tends to be an overdependence on the analyst's judgment. Different analysts with different experiences may make different predictions regarding the same problem (intra-analyst reliability). Similarly, the same analyst may make different judgments on different occasions (inter-analyst reliability). This subjectivity of analysis may weaken the confidence that can be placed in any predictions made. The analyst is required to be an expert in the technique as well as in the operation of the device being analyzed if the analysis has any potential for validity.

TABLE 8.3. Error Types and Associated Psychological Mechanisms

Error Type	Associated Psychological Mechanism
Slip	Action execution
Lapse and mode errors	Memory
Mistake	Planning and intention of action
Mistake	Interpretation and situation assessment

Two techniques are considered here because of the inherent differences in the way the methods work. Systematic Human Error Reduction and Prediction Approach (SHERPA) is a divergent error-prediction method; it works by associating up to 10 error modes with each action. In the hands of a novice, it is typical for there to be an over-inclusive strategy for selecting error modes. The novice user would rather play-safe-than-sorry and would tend to predict many more errors than would actually occur. This might be problematic; "crying wolf" too many times might ruin the credibility of the approach. Task Analysis For Error Identification (TAFEI), by contrast, is a convergent error-prediction technique; it works by identifying the possible transitions between the different states of a device and uses the normative description of behavior (provided by the Hierarchical Task Analysis (HTA)) to identify potentially erroneous actions. Even in the hands of a novice the technique seems to prevent the individual from generating too many false alarms, certainly no more than they do using heuristics. In fact, by constraining the user of TAFEI to the problem space surrounding the transitions between device states, it should exclude extraneous error prediction. Indeed, this was one of the original aims for the technique when it was originally developed (Baber & Stanton, 1994).

Systematic Human-Error Reduction and Prediction Approach (SHERPA)

SHERPA represents the error-list approach. At its core is an error taxonomy that is not unlike the classification schemes presented in the previous section. The idea is that each task can be classified into one of five basic types. SHERPA (Embrey, 1986; Stanton, 2005) uses Hierarchical Task Analysis (HTA; Annett, Duncan, Stammers, & Gray, 1971) together with an error taxonomy to identify credible errors associated with a sequence of human activity. In essence, the SHERPA technique works by indicating which error modes are credible for each task step in turn, based on an analysis of work activity. This indication is based on the judgment of the analyst, and requires input from a subject matters expert to be realistic. A summary of the procedure is shown in Fig. 8.2.

The process begins with the analysis of work activities, using Hierarchical Task Analysis. HTA (Annett et al., 1971; Annett, 2004, 2005) is based on the notion that task performance can be expressed in terms of a hierarchy of goals (what the person is seeking to achieve), operations (the activities executed to achieve the goals), and plans (the sequence in which the operations are executed). An example of HTA for the programming of a videocassette recorder is shown in Fig. 8.3

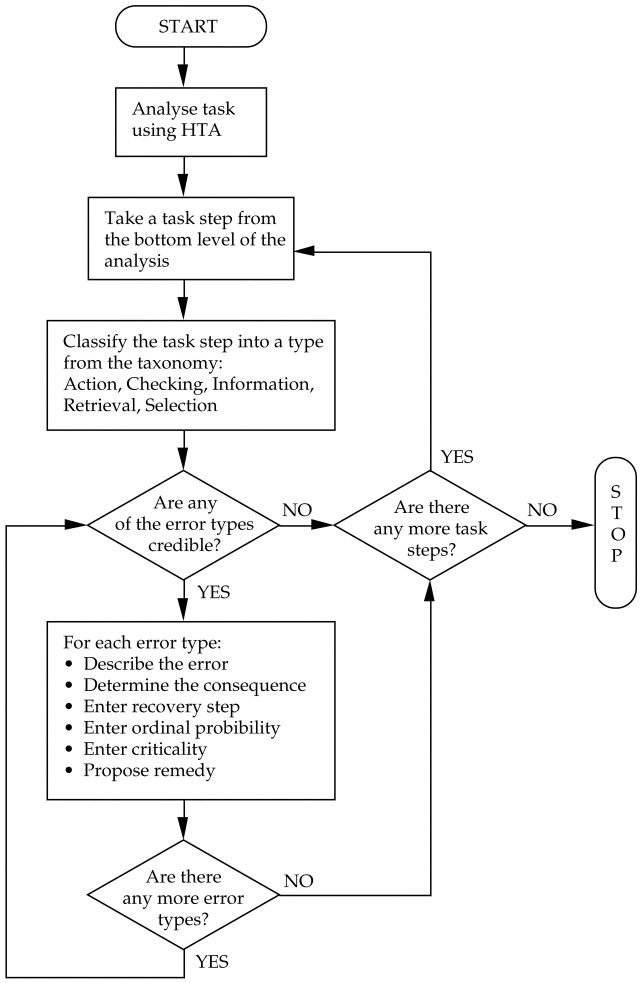


FIGURE 8.2. The SHERPA procedure.

For the application of SHERPA, each task step from the bottom level of the analysis is taken in turn. First, each task step is classified into one of the following types from the taxonomy:

- Action (e.g., pressing a button, pulling a switch, opening a door)
- Retrieval (e.g., getting information from a screen or manual)
- Checking (e.g., conducting a procedural check)
- Information communication (e.g., talking to another party)
- Selection (e.g., choosing one alternative over another)

This classification of the task step then leads the analyst to consider credible error modes associated with that activity, as shown in Table 8.4.

For each credible error (e.g., those judged by a subject matter expert to be possible) a description of the form that the error would take is given, as illustrated in Table 8.5. The consequence of the error on the system needs to be determined next, as this has implications for the criticality of the error. The last four steps consider the possibility for error recovery, the ordinal probability of the error (high, medium of low), its criticality (either critical or not critical), and potential remedies. Again, these are shown in Table 8.5.

As Table 8.5 shows, there are six basic error types associated with the activities of programming a VCR. These are,

1. Failing to check that the VCR clock is correct;
2. Failing to insert a cassette;
3. Failing to select the program number;
4. Failing to wait;
5. Failing to enter programming information correctly;
6. Failing to press the confirmatory buttons.

The purpose of SHERPA is not only to identify potential errors with the current design, but also to guide future design considerations. The structured nature of the analysis can help to focus the design remedies for solving problems, as shown in the remedial strategies column. As this analysis shows, quite a lot of improvements could be made. It is important to note, however, that the improvements are constrained by the analysis. This does not address radically different design solutions that may remove the need to program at all.

Task Analysis For Error Identification (TAFEI)

TAFEI represents the schema-based approach. It explicitly analyzes the interaction between people and machines (Baber & Stanton, 1994; Stanton & Baber, 1996; Stanton & Baber, 2005a). TAFEI analysis is concerned with task-based scenarios. This analysis is done by mapping human activity onto machine states. An overview of the procedure is shown in Fig. 8.4. TAFEI analysis consists of three principal components: Hierarchical Task Analysis (HTA); State-Space Diagrams (SSD), which are loosely based on finite state machines (Angel & Bekey, 1968); and Transition Matrices (TM). HTA provides a description of human activity, SSDs provide a description of machine activity, and TM provides a mechanism for determining potential erroneous activity through the interaction of the human and the device. In a similar manner to Newell and Simon (1972), legal and illegal operators (called "transitions" in the TAFEI methodology) are identified.

In brief, the TAFEI methodology is as follows. First, the system needs to be defined. Next, the human activities and machine states are described in separate analyses. The basic building blocks are HTA (describing human activity; see Fig. 3) and state space diagrams (describing machine activity). These two types of analysis are then combined to produce the TAFEI description of human-machine interaction, as shown in Fig. 8.5.

From the TAFEI diagram, a transition matrix is compiled and each transition is scrutinized. In Table 8.6, each transition has been classified as "impossible" (e.g., the transition cannot be performed), "illegal" (the transition can be performed but it does not lead to the desired outcome), or "legal" (the transition can be performed and is consistent with the description of error-free activity provided by the HTA), until all transitions have been analyzed. Finally, "illegal" transitions are addressed in turn as potential errors, to consider changes that may be introduced.

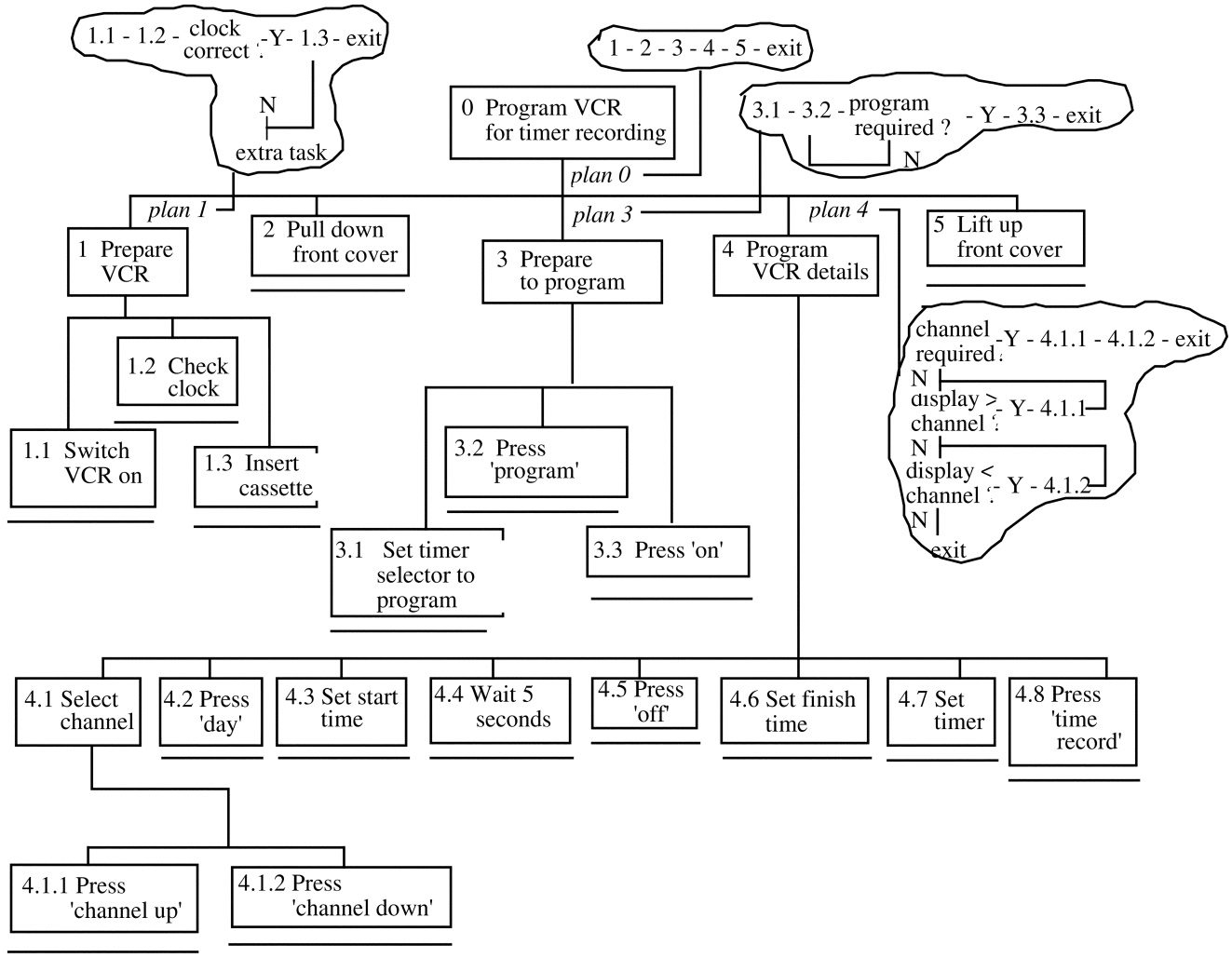


FIGURE 8.3. HTA for programming a VCR.

Thirteen of the transitions are defined as "illegal," and these can be reduced to a subset of six basic error types:

1. Switch VCR off inadvertently
2. Insert cassette into machine when switched off
3. Program without cassette inserted
4. Fail to select program number
5. Fail to wait for "on" light
6. Fail to enter programming information

In addition, one legal transition has been highlighted because it requires a recursive activity to be performed. These activities seem to be particularly prone to errors of omission. These predictions then serve as a basis for the designer to address the redesign of the VCR. A number of illegal transitions could be dealt with relatively easily by considering the use of modes in the operation of the device, such as switching off the VCR without stopping the tape and pressing play without inserting the tape. As with the SHERPA example, the point of the analysis is to help guide design efforts to make the product error-tolerant (Baber & Stanton, 2004).

VALIDATION OF HUMAN-ERROR IDENTIFICATION

There have been a few attempts to validate HEI techniques (Williams, 1989; Whalley & Kirwan, 1989; Kirwan, 1992a, 1992b; Kennedy, 1995; Baber & Stanton, 1996). For instance, Whalley and Kirwan (1989) evaluated six HEI techniques (Heuristics, PHECA, SRK, SHERPA, THERP, and HAZOP) for their ability to account for the errors known to have contributed to four genuine incidents within the nuclear industry. More recently, Kirwan (1992b) has developed a comprehensive list of eight criteria to evaluate the acceptability of these techniques at a more qualitative level. In an unpublished study, Kennedy (1995) has included Kirwan's criteria when examining the ability of the techniques to predict ten actual incidents retrospectively. While

TABLE 8.4. Error Modes and their Description

Error Mode	Error Description
Action	
A1	Operation too long/short
A2	Operation mistimed
A3	Operation in wrong direction
A4	Operation too much/little
A5	Misalign
A6	Right operation on wrong object
A7	Wrong operation on right object
A8	Operation omitted
A9	Operation incomplete
A10	Wrong operation on wrong object
Information Retrieval	
R1	Information not obtained
R2	Wrong information obtained
R3	Information retrieval incomplete
Checking	
C1	Check omitted
C2	Check incomplete
C3	Right check on wrong object
C4	Wrong check on right object
C5	Check mistimed
C6	Wrong check on wrong object
Information Communication	
I1	Information not communicated
I2	Wrong information communicated
I3	Information communication incomplete
Selection	
S1	Selection omitted
S2	Wrong selection made

these studies failed to identify a clear favorite from among these HEI techniques, all three studies indicated impressive general performance using the SHERPA method. SHERPA achieved the highest overall rankings and Kirwan (1992b) recommended a combination of expert judgment together with the SHERPA technique as the most valid approach.

The strength of these studies lies in the high level of ecological or face validity that they achieve. The methodologies make use of the opinions of expert assessors for the prediction of errors contributing to real world events. However, these studies do raise several methodological concerns. Specifically, the number of assessors using each technique is small (typically one to three), and the equivalence of assessment across techniques is brought into question because different people are assessing each HEI technique. A second methodological concern centers on the use of subjective rating scales. It is doubtful that the assessors will share the same standards when rating the acceptability or usefulness of an HEI technique. This factor, combined with the small number of assessors for each technique, means that these data should be accepted with some degree of caution.

In light of these criticisms, Baber and Stanton (1996) aimed to provide a more rigorous test for the predictive validity of SHERPA and TAFEI. Predictive validity was tested by comparing the errors identified by an expert analyst with those observed during 300 transactions with a ticket machine on the London Underground. Baber and Stanton (1996) suggested that SHERPA and TAFEI provided an acceptable level of sensitivity based on the data from two expert analysts ($d' = 0.8$). The strength of this latter study over Kirwan's was that it reports the use of the method in detail as well as the error predictions made using SHERPA and TAFEI. Stanton and Baber (2002) reported a study on the performance of SHERPA and TAFEI using a larger pool of analysts, including novice analysts, to examine the important issue of ease of acquiring the method. They reported reliability values of between 0.4 and 0.6 and sensitivity values of between 0.7 and 0.8. This compares favorably with Hollnagel, Kaarstad, and Lee's (1998) analysis of the Cognitive Reliability and Error Analysis Method (CREAM), for which they claim a 68.6% match between predicted outcomes and actual outcomes.

The research into HEI techniques suggested that the methods enable analysts to structure their judgment (Stanton & Baber, 2005b). However, the results run counter to the literature in some areas (such as usability evaluation) which suggest the superiority of heuristic approaches (Nielsen, 1993). The views of Lansdale and Ormerod (1994) may help us to reconcile these findings. They suggested that, to be applied successfully, a heuristic approach needs the support of an explicit methodology to "ensure that the evaluation is structured and thorough" (p. 257). Essentially, SHERPA and TAFEI provide a semi-structured approach which forms a framework for the judgment of the analyst without constraining it. It seems to succeed precisely because of its semi-structured nature, which alleviates the burden otherwise placed on the analyst's memory while allowing them room to use their own heuristic judgment.

APPLYING TAFEI TO INTERFACE DESIGN

A study by Baber and Stanton (1999) has shown how TAFEI can be used as part of the interface design process for a computer workstation. Their case study is based on a design project for a medical imaging software company. The software was used by cytogeneticists in research and hospital environments. The existing software was a menu-driven, text-based interface. The task of metaphase finding has six main subtasks: (a) the set-up task (where the computer is fired up, the microscope is calibrated, the scan is defined, and the cells are prepared); (b) the capture task (where the drug is applied to the cells on the slide using a pipette and the images are captured); (c) the processing task (where the background is subtracted from the cell image and the image processing is performed); (d) the analysis task (where the data are graphed and tabulated); (e) the storage task (where that images are assessed and selected images and data are saved); and (f) the shut-down task (where the computer and microscope are shut down). Using TAFEI to model the existing system, Baber and Stanton (1999) found that the sequence of activities required to conduct the metaphase-finding tasks were not supported logically by the computer interface. The two main problems were the confusing range of choices offered to users in each system state and the number of recursions in the task sequence required to perform even the sim-

TABLE 8.5. The SHERPA description.

Task Step	Error Mode	Error Description	Consequence	Recovery	P	C	Remedial Strategy
1.1	A8	Fail to switch VCR on	Cannot proceed	Immediate	L		Press any button to switch VCR on
1.2	C1	Omit to check clock	VCR Clock time may be incorrect	None	L	!	Automatic clock setting and adjust via radio transmitter
	C2	Incomplete check					
1.3	A3	Insert cassette wrong way around	Damage to VCR	Immediate	L	!	Strengthen mechanism On-screen prompt
	A8	Fail to insert cassette	Cannot record	Task 3	L		
2	A8	Fail to pull down front cover	Cannot proceed	Immediate	L		Remove cover to programming
3.1	S1	Fail move timer selector	Cannot proceed	Immediate	L		Separate timer selector from programming function
3.2	A8	Fail to press PROGRAM	Cannot proceed	Immediate	L		Remove this task step from sequence
3.3	A8	Fail to press ON button	Cannot proceed	Immediate	L		Label button START TIME
4.1.1	A8	Fail to press UP button	Wrong channel selected	None	M	!	Enter channel number directly from keypad
4.1.2	A8	Fail to press DOWN button	Wrong channel selected	None	M	!	Enter channel number directly from keypad
4.2	A8	Fail to press DAY button	Wrong day selected	None	M	!	Present day via a calendar
4.3	I1	No time entered	No program recorded	None	L	!	Dial time in via analogue clock
	I2	Wrong time entered	Wrong program recorded	None	L	!	Dial time in via analogue clock
4.4	A1	Fail to wait	Start time not set	Task 4.5	L		Remove need to wait
4.5	A8	Fail to press OFF button	Cannot set finish time				Label button FINISH TIME
4.6	I1	No time entered	No program recorded	None	L	!	Dial time in via analogue clock
	I2	Wrong time entered	Wrong program recorded	None	L	!	Dial time in via analogue clock
4.7	A8	Fail to set timer	No program recorded	None	L	!	Seperate timer selector from programming function
4.8	A8	Fail to press TIME RECORD button	No program recorded	None	L	!	Remove this task step from sequence
5	A8	Fail to lift up front cover	Cover left down	Immediate	L		Remove cover to programming

plest of tasks. Interviews with the cytogeneticists revealed some fundamental problems with the existing software, such as: (a) the existing system was not error tolerant, (b) there was a lack of consistency between commands, (c) the error and feedback messages were not meaningful to users, and (d) the same data had to be entered several times in the procedure. The lack of trust that the users had in the system meant that they also kept a paper-based log of the information they entered into the computer, meaning even greater duplication of effort.

From their analysis of the tasks sequences, Baber and Stanton (1999) developed a TAFEI diagram of the ideal system states that would lead the user through the metaphase-finding task in a logical manner. One of the software engineers described the analysis as "a sort of video playback of someone actually doing things with the equipment." The TAFEI analysis was a revelation to the company, who had some vague idea that all was not well with the interface, but had not conceived of the problem as defining the task sequence and user actions. As a modeling approach, TAFEI does this rather well. The final step in the analysis was to define and refine the interface screen. This was based on the task sequence and TAFEI description, to produce a prototype layout for comment with the users and software engineers. Specific task scenarios were then tried out to see if the new interface would support user activity. Following some minor modification, the company produced a functional prototype for user performance trials.

Baber and Stanton (1999) reported successful completion of the interface design project using TAFEI. They argued that the method supports analytical prototyping activity. TAFEI enables designers to focus attention on the task sequence, user activity, and interface design. It also highlights potential problem points in the interaction, where errors are likely to occur. While it is accepted that these considerations might also form part of other methods (e.g., storyboards), by contrast TAFEI does this in a structured, systematic rigorous and auditable approach for the consideration of human error potential in system design.

APPLYING SHERPA TO SAFETY CRITICAL SYSTEMS

Studies reported on SHERPA show how it can be applied to the evaluation of energy distribution (Glendon & Stanton, 1999) and oil extraction (Stanton & Wilson, 2000) in the safety-critical industries. Both of these represent multiperson systems. The study by Glendon and Stanton (1999) compared the impact of the implementation of a new safety management system on error potential in an energy-distribution company. The first assessment of human-error potential in electrical switching operations was undertaken in 1994. Following this, the company undertook major organization-restructuring activities. This in-

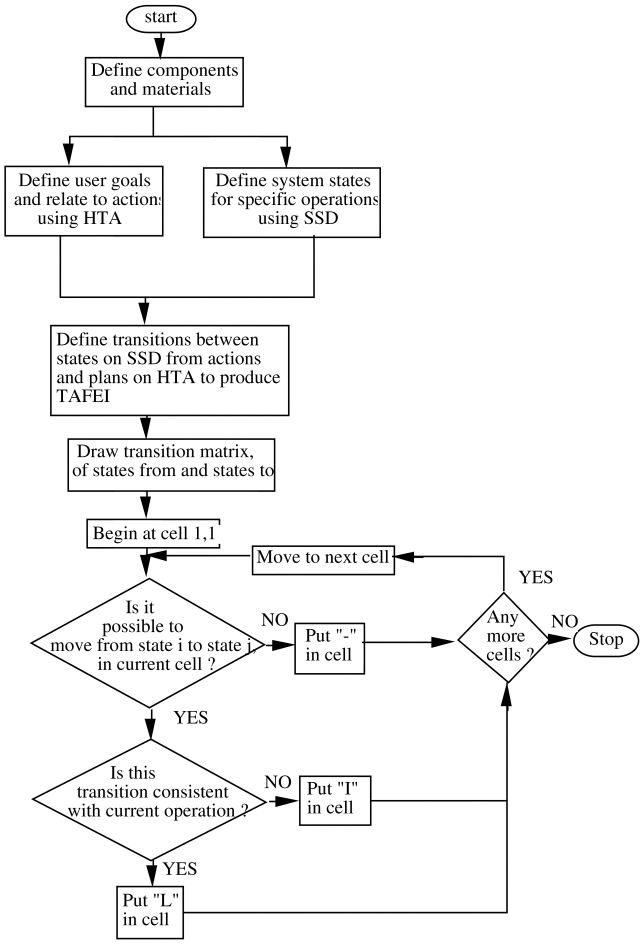


FIGURE 8.4. The TAFEI procedure.

cluded a major safety-management initiative, which was to separate operational control of the energy distribution system from safety management. In the original assessment of the system in 1994 it became obvious that safety was often secondary to operational performance. In order to remove this conflict, a separate safety-management center was established. This enabled the operational center to pass control of part of the energy-distribution system over to the safety-management center when safety issues arose. Thus operational control and safety management were not in conflict. When the safety issues were resolved, control of the energy-distribution system was passed back to the operational center. The effect on reducing error potential in the system is illustrated in Fig. 8.6.

Statistical analyses, using the Binomial test, were conducted to determine the difference between the error rates in 1994 with those in 1997. All comparisons between critical (low and medium likelihood) errors proved statistically significant ($p < 0.001$). It should be noted that there were no "high" likelihood errors in the analyses of this system, all of which had probably been designed out of the system as it evolved. Fig. 8.6 shows a reduction in the error potential by over 50% for critical errors from 1994 to 1997. This suggests that the system is likely to be safer. The SHERPA technique assisted in the redesign process after the 1994 analysis by proposing design-reduction strategies. The results of these proposals are seen in the 1997 analysis.

Stanton and Wilson (2000) described the assessment of an oil drilling task undertaken offshore using the SHERPA technique. Surprisingly, the oil industry lacked a method for assessing the robustness of its working practices. SHERPA scrutinized the minutiae of human action, and provided some basis for assessing the magnitude of increased risk from human activity, which could then be used to inform which intervention strategies were likely to make the working environment safer. Drilling is a highly integrated team activity comprised of a site controller, drilling supervisor, geologist, toolpusher, directional driller, assistant driller, mudlogger, mud engineer, derrickman, crane operator, roughnecks, and roustabouts. It is also a fairly dangerous occupation, as the incidents at Piper Alpha and Ocean Odyssey show. It is argued that the risks arising from human operations can be controlled in the same way as engineering risk. Stanton and Wilson (2000) conducted a detailed task analysis of drilling operations, covering monitoring the status of the well (monitor), detecting abnormalities (detect), and dealing with kicks (deal). The results of the SHERPA analysis are shown in Fig. 8.7.

As Fig. 8.7 shows, the error potential in the detection of abnormalities task and the dealing with kicks task are considerably greater than the monitor the well task. This should be some cause for concern, and certainly explains some of the problems that oil companies have observed. The proposed remedies from the SHERPA analysis included strategies such as computer-prompted checks of drilling variables, placement of drilling parameters on separate displays, redesign of trend displays, computer-prompted alarm levels, automatic transmission of information, electronic links between the drilling team members, computer-generated tables of mud loss, computer-based procedures, online weather systems, and automated shutdown procedures. All of these proposals are really about the design of the human-computer interface. Stanton and Wilson (2000) argued that adopting these approaches should help organizations design safer systems and help prevent catastrophic disasters in the future.

CONCLUSIONS

Despite most HEI techniques, and ergonomics methods resulting from them being used in an evaluative and summative manner, it is entirely feasible for them to be used in a formative design approach. Stanton and Young (1999) argued that the methods may have the greatest impact at the prototyping stage, particularly considering one of the key design stages: *analytic prototyping*. Although in the past it may have been costly, or even impossible, to alter design at the structural prototyping stage, with the advent of computer-aided design it is made much simpler. It may even be possible to compare alternative designs at this stage with such technology. In terms of the analytical prototyping of human interfaces, Baber and Stanton (1999) argued that there are three main forms: functional analysis (e.g., consideration of the range of functions the device sup-

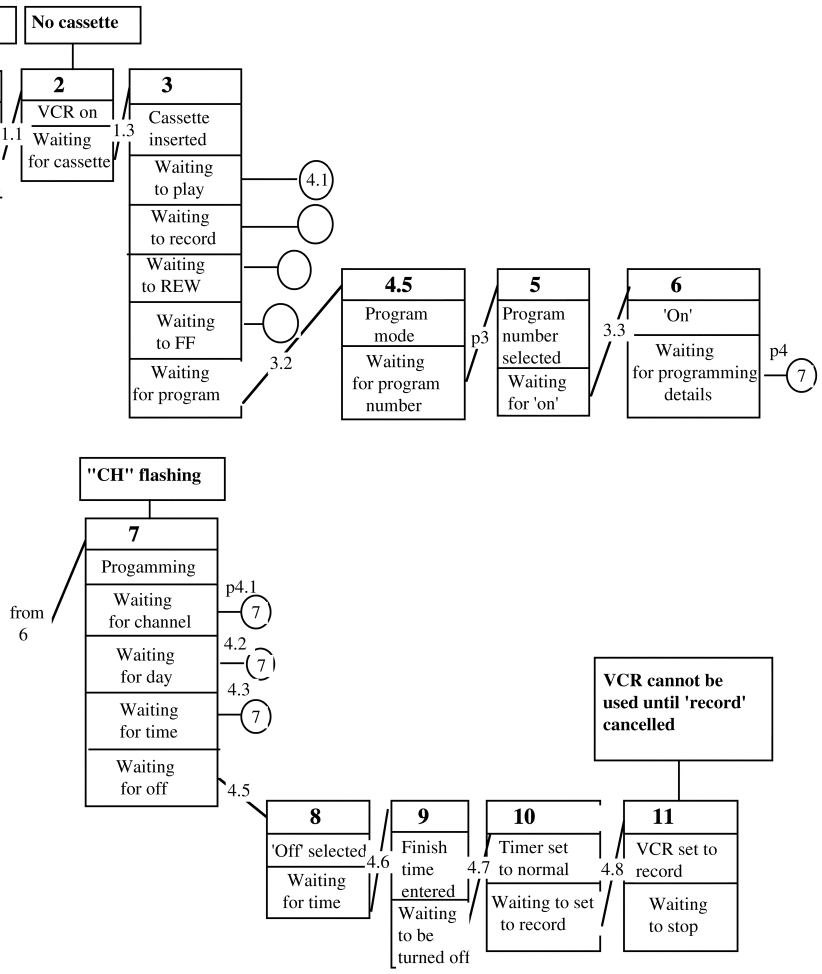


FIGURE 8.5. The TAFEI description.

TABLE 8.6. The Transition Matrix

From state: \ To state:	1	2	3	4.5	5	6	7	8	9	10	11
1	-	L	I	-	-	-	-	-	-	-	-
2	L	-	L	I	-	-	-	-	-	-	-
3	L	-	-	L	-	-	-	-	-	-	-
4.5	I	-	-	-	L	I	-	-	-	-	-
5	I	-	-	-	-	L	I	I	-	-	-
6	I	-	-	-	-	-	L	I	-	-	-
7	I	-	-	-	-	-	L	L	-	-	-
8	I	-	-	-	-	-	-	-	L	-	-
9	I	-	-	-	-	-	-	-	-	L	-
10	I	-	-	-	-	-	-	-	-	-	L
11	-	-	-	-	-	-	-	-	-	-	-

ports), scenario analysis (e.g., consideration of the device with regard to a particular sequence of activities), and structural analysis (e.g., non-destructive testing of the interface from a user-centered perspective). The case studies of how TAFEI and SHERPA were applied on interface design projects demonstrate how system improvements may be made. Stanton and Young (1999) have extended this analysis to other ergonomics methods. There are three take-home messages from this chapter, which are that

1. Most technology-induced errors are entirely predictable;
2. Structured methods, such as SHERPA and TAFEI, produce reliable and valid error data;
3. Ergonomics methods should be used as part of the formative design process, to improve design and reduce errors.

Successful design will require expertise in the subject of analysis as well as expertise in the methodology being used. Exploring design weaknesses through SHERPA and TAFEI will help in developing more error-tolerant devices and products.

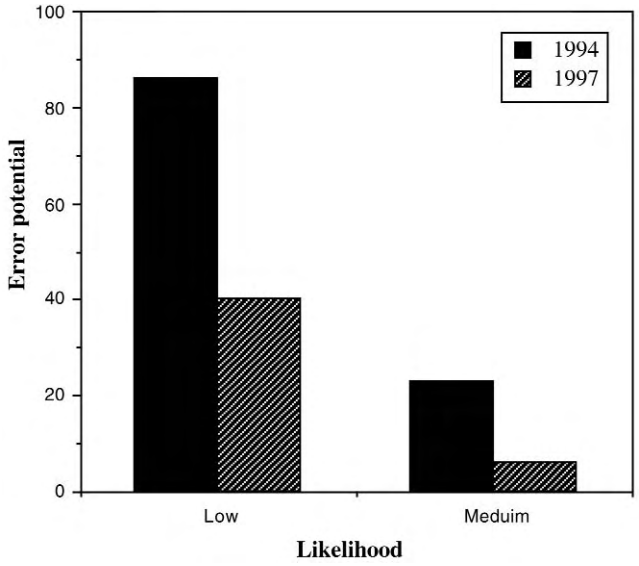


FIGURE 8.6. SHERPA analysis of an energy-distribution system.

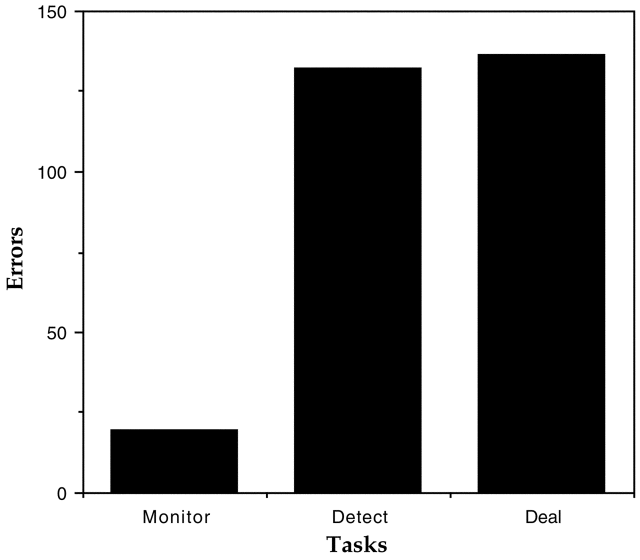


FIGURE 8.7. SHERPA analysis of drilling for oil tasks.

References

Annett, J. (2004). Hierarchical task analysis. In D. Diaper, & N. A. Stanton (Eds.), *The handbook of task analysis for human-computer iInteraction* (pp. 67–82). Mahwah, New Jersey: Lawrence Erlbaum.

Annett, J. (2005). Hierarchical task analysis. In N. A. Stanton, A. Hedge, E. Salas, H. Hendrick, & K. Brookhaus (Eds.), *Handbook of human factors and ergonomics methods* (pp. 33-1–33-7). London: Taylor & Francis,.

Annett, J., Duncan, K. D., Stammers, R. B., & Gray, M. J. (1971). *Task analysis. Training information, No. 6.* London: HMSO.

Angel, E. S., & Bekey, G. A. (1968). Adaptive finite state models of manual control systems. *IEEE Transactions on Man-Machine Systems*, March, 15–29.

Baber, C., & Stanton, N. A. (1994). Task analysis for error identification: a methodology for designing error tolerant consumer products. *Ergonomics, 37*(11), 1923–1941.

Baber, C., & Stanton, N. A. (1996). Human error identification techniques applied to public technology: Predictions compared with observed use. *Applied Ergonomics, 27*(2), 119–131.

Baber, C., & Stanton, N. A. (1999). Analytical prototyping. In J. M. Noyes, & M. Cook (Eds.), *Interface technology: The leading edge* (pp. 175–194). Baldock, UK: Research Studies Press..

Baber, C., & Stanton, N. A. (2004). Task analysis for error identification. In D. Diaper, & N. A. Stanton (Eds.), *The handbook of task analysis for human-computer interaction* (pp. 367–379). Mahwah, New Jersey: Lawrence Erlbaum Associates.,

Casey, S. (1993). *Set Phasers on Stun and other true tales of design, technology and human error.* Santa Barbara, California: Aegean.

Chapanis, A. (1999). *The Chapanis Chronicles: 50 years of human factors research, education, and design.* Santa Barbara, California: Aegean.

Embrey, D. E. (1986). *SHERPA: A Systematic Human Error Reduction and Prediction Approach*. Paper presented at the International Meeting on Advances in Nuclear Power Systems, Knoxville, Tennessee.

Glendon, A. I., & Stanton, N. A. (1999). Safety culture in organisations. *Safety Science, 34,* 193–214.

Hollnagel, E. (1993). *Human Reliability Analysis: context and control.* London: Academic Press.

Hollnagel, E., Kaarstad, M., & Lee, H. C. (1998). Error mode prediction. *Ergonomics, 42*, 1457–1471.

Kennedy, R. J. (1995). Can human reliability assessment (HRA) predict real accidents? A case study analysis of HRA. In A. I. Glendon, & N. A. Stanton (Eds.), *Proceedings of the risk assessment and risk reduction conference,* Birmingham: Aston University..

Kirwan, B. (1992a). Human error identification in human reliability assessment. Part 1: overview of approaches. *Applied Ergonomics, 23*, 299–318.

Kirwan, B. (1992b). Human error identification in human reliability assessment. Part 2: detailed comparison of techniques. *Applied Ergonomics, 23*, 371–381.

Lansdale, M. W., & Ormerod, T. C. (1994). *Understanding interfaces*. London: Academic Press.

Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology.* San Francisco: Freeman.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Nielsen, J. (1993). *Usability Engineering*. Boston: Academic Press.

Norman, D. A. (1981). Categorisation of action slips. *Psychological Review, 88*(1) 1–15.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic Books.

Reason, J. (1990). *Human error.* Cambridge, UK: Cambridge University Press.

Senders, J. W., & Moray, N. P. (1991). *Human error.* Hillsdale, NJ: LEA.

Stanton, N. A. (2005). Systematic human error reduction and prediction approach. In N. A. Stanton, A. Hedge, E. Salas, H. Hendrick, & K. Brookhaus (Eds.), *Handbook of human factors and ergonomics methods* (pp. 371–378). London: Taylor & Francis.

Stanton, N. A., & Baber, C. (1996). A systems approach to human error identification. *Safety science, 22*, 215–228.

Stanton, N. A., & Baber, C. (2002). Error by design: Methods for predicting device usability. *Design Studies, 23*, 363–384.

Stanton, N. A., & Baber, C. (2005a). Task analysis for error identification. In N. A. Stanton, A. Hedge, E. Salas, H. Hendrick, & K. Brookhaus (Eds.), *Handbook of human factors and ergonomics methods* (pp. 381–389). London: Taylor & Francis.

Stanton, N. A., & Baber, C. (2005b). Validating task analysis for error identification: Reliability and validity of a human error prediction technique. *Ergonomics, 48*, 1097–1113.

Stanton, N. A., & Wilson, J. (2000). Human factors: Step change improvements in effectiveness and safety. *Drilling Contractor,* Jan/Feb, 46–51.

Stanton, N. A., & Young, M. S. (1999). *A guide to methodology in ergonomics.* London: Taylor and Francis.

Wagenaar, W. A., & Groeneweg, J. (1988). Accidents at sea: Multiple causes, impossible consequences. *International Journal of Man-Machine Studies, 27*, 587–598.

Whalley, S. J., & Kirwan, B. (1989, June). *An evaluation of five human error identification techniques*. Paper presented at the 5th International Loss Prevention Symposium, Oslo, Norway.

Wickens, C. D. (1992). *Engineering psychology and human performance*. New York: Harper Collins.

Williams, J. C. (1989). Validation of human reliability assessment techniques. *Reliability Engineering, 11*, 149–162.

Part

II

COMPUTERS IN HCI

•9•

INPUT TECHNOLOGIES AND TECHNIQUES

Ken Hinckley
Microsoft Research

What's An Input Device Anyway? **162**
Understanding Input Technologies **162**
Input Device Properties 162
Property sensed 162
Number of dimensions 162
Indirect versus direct 162
Device acquisition time 162
Gain 162
Other metrics 162
A Brief Tour of Pointing Devices 163
Mice 163
Trackballs 163
Isometric joysticks 163
Isotonic joysticks 163
Indirect tablets 163
Touchpads 163
Touchscreens and pen-operated devices 163
Input Device States 163
What's an Input Device For? The Composition of User Tasks **165**
Elemental Tasks 165
Compound Tasks and Chunking 165
Evaluation and Analysis of Input Devices **165**
Fitts' Law and Hick's Law 166
The Steering Law and Minimum Jerk Law 166
The Keystroke-Level Model (KLM) and GOMS Analysis 166
Transfer Functions: How to Transform an Input Signal **167**
Self-centering devices 167
Motion-sensing devices 167
Absolute devices 167
Feedback: What Happens in Response to an Input? . . **167**
Proprioceptive and Kinesthetic Feedback 167
Kinesthetic Correspondence 167
Snapping Behaviors and Active Haptic Feedback 168
Keyboards and Text Entry **168**
Procedural Memory 168
Mobile Text Entry, Character Recognition, and Handwriting Recognition 168
Modalities of Interaction **169**
Speech and Multimodal Input 169
Bimanual Input 169
Pen and Gesture Input 169
Whole Hand Input 170
Background Sensing Techniques 170
Multitouch Tables and Screens 171
A Society of Devices 171
Current and Future Trends for Input **171**
References **172**

WHAT'S AN INPUT DEVICE ANYWAY?

Input devices sense physical properties of people, places, or things. Yet any treatment of input devices without regard to the corresponding visual feedback is like trying to use a pen without paper. Small-screen devices with integrated sensors underscore the indivisibility of input and output. This chapter treats input technologies at the level of *interaction techniques*, which provide a way for users to accomplish tasks by combining input with appropriate feedback. An interaction designer must consider (a) the physical sensor, (b) the feedback presented to the user, (c) the ergonomic and industrial design of the device, and (d) the interplay between all of the interaction techniques supported by a system.

This chapter enumerates properties of input devices and provides examples of how these properties apply to common pointing devices, as well as mobile devices with touch or pen input. We will discuss how to use input signals in applications, as well as models and theories that help to evaluate interaction techniques and reason about design options. We will also discuss discrete symbolic entry, including mobile and keyboard-based text entry. The chapter concludes with some thoughts about future trends.

UNDERSTANDING INPUT TECHNOLOGIES

A designer who understands input technologies and the task requirements of users has a better chance of designing interaction techniques that match a user's natural workflow. Making an optimal choice for tasks in isolation leads to a poor design, so the designer must weigh competing design requirements, as well as transitions between tasks.

Input Device Properties

The variety of pointing devices is bewildering, but a few important properties characterize most input sensors. These properties help a designer understand a device and anticipate potential problems. We will first consider these device properties in general, and then show how they apply to some common input devices.

Property sensed. Most devices sense linear position, motion, or force; rotary devices sense angle, change in angle, and torque (Buxton, 1995c; Card, Mackinlay, & Robertson, 1991). For example, tablets sense position of a pen, mice sense motion (change in position), and isometric joysticks sense force. The property sensed determines the mapping from input to output, or *transfer function*, that is most appropriate for the device. Position sensing devices are *absolute input devices*, whereas motion-sensing devices are *relative input devices*. A relative device, such as the mouse, requires visual feedback in the form of a cursor to indicate a screen location. With absolute devices, the *nulling problem* (Buxton, 1983) arises if the position of a physical intermediary, such as a slider on a mixing console, is not in agreement with a value set in software. This problem cannot occur with relative devices, but users may waste time *clutching*: the user must occasionally lift a mouse to reposition it.

Number of dimensions. Devices sense one or more input dimensions. For example, a mouse senses two linear dimensions of motion, a knob senses one angular dimension, and a six degree-of-freedom magnetic tracker measures three position dimensions and three orientation dimensions. A pair of knobs or a mouse with a scroll wheel sense separate input dimensions and thus form a "1D + 1D" device, or a "2D + 1D" *multichannel device*, respectively (Zhai, Smith, & Selker, 1997). *Multidegree-of-freedom devices (3D input devices)* sense three or more simultaneous dimensions of spatial position or orientation (Bowman, Kruijff, LaViola, & Poupyrev, 2004; Hinckley, Pausch, Goble, & Kassell, 1994; Hinckley, Sinclair, Hanson, Szeliski, & Conway, 1999; Zhai, 1998).

Indirect versus direct. A mouse is an indirect input device because the user must move the mouse to indicate a point on the screen, whereas a direct input device has a unified input and display surface. Direct devices such as touchscreens, or display tablets operated with a pen, are not necessarily easier to use than indirect devices. Direct devices lack buttons for state transitions. Occlusion is also a major design challenge. The finger or pen occludes the area at which a user is pointing, so the user may not realize that he or she has activated a control; occlusion by the hand and arm also may cause the user to overlook pop-up menus, dialogs, or status indicators.

Device acquisition time. The average time to move one's hand to a device is known as *acquisition time*. *Homing time* is the time to return from a device to a "home" position (e.g., return from mouse to keyboard). For common desktop workflows that involve switching between text entry and pointing, the effectiveness of a device for pointing tends to dominate acquisition and homing time costs (Douglas & Mithal, 1994). Thus, integration of a pointing device with the keyboard may not improve overall performance, but evaluations still must assess any influence of acquisition times (Dillon, Eday, & Tombaugh, 1990; Hinckley et al., 2006).

Gain. Also known as control-to-display (C:D) gain or C:D ratio, *gain* is defined as the distance moved by an input device divided by the distance moved on the display. Gain confounds what should be two measurements—(a) device size and (b) display size—with one arbitrary metric (Accot & Zhai, 2001; MacKenzie, 1995), and is therefore suspect as a factor for experimental study. In experiments, gain typically has little or no effect on the time to perform pointing movements, but variable gain functions may provide some benefit by reducing the required *footprint* (physical movement space) of a device.

Other metrics. System designers must weigh other performance metrics, including pointing speed and accuracy, error rates, learning time, footprint, user preference, comfort, and cost (Card, Mackinlay, & Robertson, 1990). Other important engineering parameters include sampling rate, resolution, accuracy, and linearity (MacKenzie, 1995).

A Brief Tour of Pointing Devices

Most operating systems treat all input devices as *virtual devices*, which tempts one to believe that pointing devices are interchangeable; however, the details of what the input device senses, such as how it is held, the presence or absence of buttons, and many other properties, can significantly impact the interaction techniques—and hence, the end-user tasks—that a device can effectively support. The following tour discusses important properties of several common pointing devices.

Mice. Douglas Englebart and colleagues (English, Englebart, & Berman, 1967) invented the mouse in 1967 at the Stanford Research Institute. Forty years later, the mouse persists because its properties provide a good match between human performance and the demands of graphical interfaces (Balakrishnan, Baudel, Kurtenbach, & Fitzmaurice, 1997). For typical pointing tasks on a desktop computer, one can point with the mouse about as well as with the hand itself (Card, English, & Burr, 1978). Because the mouse stays put when the user releases it (unlike a stylus, for example), it is quick for users to reacquire and allows designers to integrate multiple buttons or other controls on its surface. Users exert force on mouse buttons in a direction orthogonal to the mouse's plane of motion, thus minimizing inadvertent motion. Finally, with mice, all of the muscle groups of the hand, wrist, arm, and shoulder contribute to pointing, allowing high performance for both rapid, coarse movements as well as slow, precise movements (Guiard, 1987; Zhai, Milgram, & Buxton, 1996). These advantages suggest the mouse is hard to beat; it will remain the pointing device of choice for desktop graphical interfaces.

Trackballs. A trackball senses the relative motion of a partially exposed ball in two degrees of freedom. Trackballs have a small working space (*footprint*), and afford use on an angled surface. Trackballs may require frequent clutching movements because users must lift and reposition their hand after rolling the ball a short distance. The buttons are located to the side of the ball, which can make them awkward to hold while rolling the ball (MacKenzie, Sellen, & Buxton, 1991). A trackball engages different muscle groups than a mouse, offering an alternative for users who experience discomfort when using a mouse.

Isometric joysticks. An isometric joystick (e.g., the IBM Trackpoint) is a force-sensing joystick that returns to center when released. Most isometric joysticks are stiff, offering little feedback of the joystick's displacement. The rate of cursor movement is proportional to the force exerted on the stick; as a result, users must practice in order to achieve good cursor control. Isometric joysticks may offer the only pointing option when space is at a premium (Douglas & Mithal, 1994; Rutledge & Selker, 1990; Zhai et al., 1997).

Isotonic joysticks. Isotonic joysticks sense angle of deflection. Some hybrid designs blur the distinctions of isometric and isotonic joysticks, but the main questions are, "Does the joystick sense force or angular deflection?" "Does the stick return to center when released?" and "Does the stick move from the starting position?" For a discussion of the complex design space of joysticks, see Lipscomb and Pique (1993).

Indirect tablets. Indirect tablets report the absolute position of a pointer on a sensing surface. *Touch tablets* sense a bare finger, whereas *graphics tablets* or *digitizing tablets* typically sense a stylus or other physical intermediary. Tablets can operate in *absolute mode*, with a fixed C:D gain between the tablet surface and the display, or in *relative mode*, in which the tablet responds only to motion of the stylus. If the user touches the stylus to the tablet in relative mode, the cursor resumes motion from its previous position; in absolute mode, it would jump to the new position. Absolute mode is generally preferable for tasks such as drawing, handwriting, tracing, or digitizing, but relative mode may be preferable for typical desktop interaction tasks such as selecting graphical icons or navigating through menus. Tablets thus allow coverage of many tasks (Buxton, Hill, & Rowley, 1985), whereas mice *only* operate in relative mode.

Touchpads. Touchpads are small, touch-sensitive tablets often found on laptop computers. Touchpads use relative mode for cursor control because they are too small to map to an entire screen, but most touchpads also have an absolute mode to allow features such as sliding along the edge of the pad to scroll. Touchpads support clicking by recognizing tapping or double-tapping gestures, but accidental contact (or loss of contact) can erroneously trigger such gestures (MacKenzie & Oniszczak, 1998). Like trackballs, the small size of touchpads necessitates frequent clutching, and touchpads can be awkward to use while holding down a button, unless the user employs his or her other hand.

Touchscreens and pen-operated devices. Touchscreens are transparent, touch-sensitive tablets mounted on a display. Some touchscreens can only sense a bare finger; others can sense either a plastic stylus or a bare finger. Transparent electromagnetic digitizers, such as those found on the Tablet PC, cannot sense touch, and require the use of a special pen. *Parallax error* is a mismatch between the sensed input position and the apparent input position due to viewing angle; look to minimize the displacement between the sensing and display surfaces to avoid this problem. Depending on the mounting angle, a touch or pen-operated display may result in arm or neck fatigue (Sears, Plaisant, & Shneiderman, 1992). A touchscreen that is integrated with a mobile device can be prone to accidental contact when the user picks up the device. Yet mobile devices that only sense a pen require the user to unsheathe the stylus to perform any interaction; the user cannot quickly poke at the screen with a finger. The limited states and events sensed by pen or touch-operated devices raise additional design challenges, as discussed below.

Input Device States

There is a fundamental mismatch between the demands of graphical interfaces and the states and events that can be sensed by devices such as touchscreens and pen-operated devices, which makes it difficult to support the full set of graphical in-

terface primitives, including click, drag, double-click, and right-click. There is no easy solution when using nonstandard pointing devices that does not involve design compromises. When considering such devices, to make device limitations and differences concrete, one of the first things a designer should do is diagram all of these states and transitions.

Input devices taken in general support three possible states (Fig. 9.1): (a) out-of-range, (b) tracking, and (c) dragging; practitioners refer to these as State 0, State 1, and State 2, respectively, of the three-state model (Buxton, 1990). This model is useful for reasoning about the relationship between the events sensed by an input device and the demands of interaction techniques.

The three-state model describes the mouse as a two-state device, supporting State 1, the cursor tracking state, as well as State 2, the dragging state. State 1 provides cursor feedback of the screen position that the device will act upon, while State 2 allows the user to drag an object by holding down the primary mouse button while moving the mouse. The mouse senses movement in both the tracking and dragging states, as represented by the *dx*, *dy* in each state (Fig. 9.2, left), indicating relative motion tracking capability.

Many touch-activated devices such as touchscreens, touchpads, and PDA screens are also two-state devices, but *do not sense the same two states as the mouse* (Fig. 9.2, right). For example, a PDA can sense a finger when it is in contact with the screen; this is the equivalent of the mouse dragging state (State 2). The PDA can also sense when the finger is removed from the screen, but once the finger breaks contact, this enters State 0 (out-of-range), where no motion can be detected (emphasized by the *nil* in state 0 of Fig. 9.2, right). Thus, although the mouse and PDA screen both sense two states, the lack of a second motion-sensing state on the PDA means that it will be difficult to support the same interaction techniques as a mouse. For example, should sliding one's finger on the screen move a cursor, or drag an object? The designer must choose one; the PDA screen cannot support both behaviors at the same time.

The Tablet PC is an example of a pen-operated device that senses all three states of the three-state model (Fig. 9.3). The Tablet PC senses the location of the stylus when it is proximate to the screen. The pen triggers an event when it makes contact with the screen, as well as when it enters or leaves proximity.

Unfortunately, even with all three states, it is still difficult for a pen to support all the interaction techniques offered by a mouse. To help illustrate why this is the case, we can extend the three-state model to more fully characterize the interaction techniques at the core of graphical user interfaces (Fig. 9.4). The resulting five states of graphical user interfaces are (a) Tracking (1), (b) Hover (1_H), (c) Left Click (2_L), (d) Dragging (2_D), and (e) Right Click (2_R).

This diagram suggests will be difficult to support all of these interactions on a pen or touch-based device; there is no elegant solution in the literature. In this five-state model, a click is the series of transitions *1-2$_L$-1* with no motion in state 2_L, and a double click is *1-2$_L$-1-2$_L$-1* with no motion between the two

State	Description
0	*Out Of Range*: The device is not in its physical tracking range.
1	*Tracking*: Device motion moves only the cursor.
2	*Dragging*: Device motion moves objects on the screen.

FIGURE 9.1. Summary of states in Buxton's three-state model. Adapted from (Buxton, 1990).

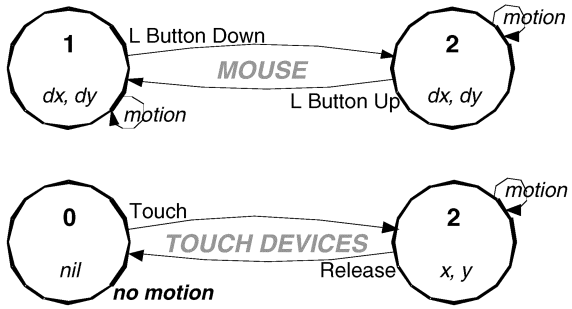


FIGURE 9.2. States sensed by a mouse (left) versus states sensed by touch-operated devices such as touchpads (right). Adapted from (Hinckley, Czerwinski, & Sinclair, 1998a)

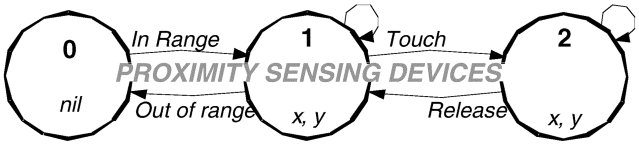


FIGURE 9.3. States sensed by a Tablet PC pen. Adapted from (Hinckley et al., 1998a).

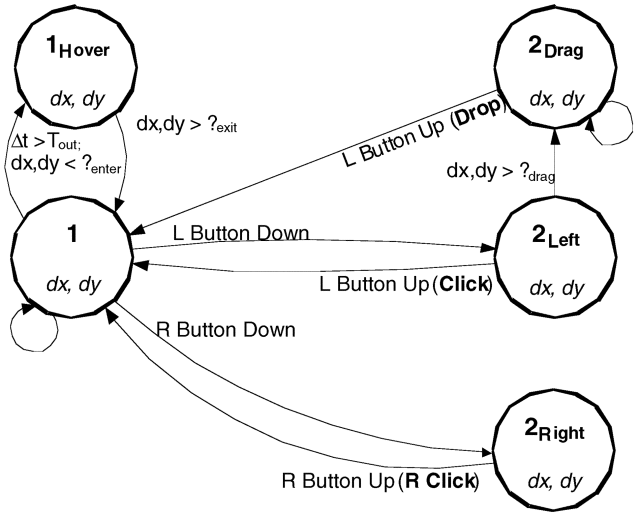


FIGURE 9.4. The five states and transition rules at the core of graphical user interfaces.

clicks. Even if the device can sense State 1, hovering (1_H) for help text requires holding the pointer motionless above the display. These gestures are all difficult to perform with a pen or finger because touching or lifting disturb the cursor position (Buxton et al., 1985). Furthermore, because the pen may be out of range, and because the user must move the pen through the tracking zone to enter State 2, pen operated input devices lack a well-defined current cursor position.

Pen and touch devices also lack a second button for right-click. A finger obviously has no buttons, but even a barrel button on a pen is slow to access and is prone to inadvertent activation (Li, Hinckley, Guan, & Landay, 2005). Some mobile devices use dwelling with the pen or finger as a way to simulate right-click, but the timeout introduces an unavoidable delay; for rapid activation, it should be short as possible, but to avoid inadvertent activation (e.g., if the user is thinking about the next action to take while resting the pen on the screen), the timeout must be as long as possible. A 500-millisecond timeout offers a reasonable compromise (Hinckley, Baudisch, Ramos, & Guimbretiere, 2005a). Even techniques designed for pen-operated devices (Apitz & Guimbretiere, 2004; Kurtenbach & Buxton, 1991; Moran, Chiu, & van Melle, 1997) require rapid and unambiguous activation of one of several possible actions as fundamental building blocks; otherwise, inefficient or highly modal interactions become necessary and may reduce the appeal of such devices (Hinckley et al., 2006).

Similar issues plague other interaction modalities, such as (a) motion-sensing mobile devices (Hinckley, Pierce, Horvitz, & Sinclair, 2005b), (b) camera-based tracking of the hands (Wilson & Oliver, 2003), and (c) 3D input devices (Hinckley et al., 1994). All of these techniques require a method for users to move the device or their hands without accidentally performing an action. Thus state transitions form fundamental indications of intent that are essential for rapid and dependable interaction.

WHAT'S AN INPUT DEVICE FOR? THE COMPOSITION OF USER TASKS

One way of reasoning about input devices and interaction techniques is to view a device or technique in light of the tasks that it can express. But what sort of tasks are there?

Elemental Tasks

While computers can support many activities, at the input level, some subtasks appear repeatedly in graphical user interfaces, such as pointing at a target on the screen or typing a character. Foley and colleagues (Foley, Wallace, & Chan, 1984) identified elemental tasks including (a) *text* (entering symbolic data), (b) *select* (indicating objects from a set of alternatives), (c) *position* (pointing to a screen coordinate), and (d) *quantify* (specifying an exact numeric value). If these are elemental tasks, however, then where do devices such as global positioning system (GPS) readers, cameras, or fingerprint scanners fit in? These offer new "elemental" data types (e.g., location, images, and identity). Advances in technology will continue to yield data types that enable new tasks and scenarios of use.

Compound Tasks and Chunking

Another problem with the elemental task approach is that the level of analysis for "elemental" tasks is not well defined. For example, a mouse indicates an *(x, y)* position on the screen, but an Etch A Sketch® separates positioning into two subtasks by providing a single knob for *x* and a single knob for *y* (Buxton, 1986b). If *position* is an elemental task, why must we subdivide this task for some devices but not others? One way to resolve this puzzle is to view all tasks as hierarchies of subtasks (Fig. 9.5). Whether or not a task is "elemental" depends on the input device being used: the Etch A Sketch supports separate *QuantifyX* and *QuantifyY* tasks, whereas the mouse supports a compound *2D Position* task (Buxton, 1986a).

From the user's perspective, a series of elemental tasks may seem like a single task. For example, scrolling a web page to click on a link could be conceived as an elemental 1D positioning task followed by a 2D selection task, or, it can be viewed as a compound *navigation/selection* task (Buxton & Myers, 1986). An interaction technique can encourage the user to work at the higher level of the compound task, for example, by scrolling with one hand while pointing to the link with the other hand. This is known as *chunking*.

These examples show that the choice of device influences the level at which the user is required to think about the individual actions that must be performed to achieve a goal. The design of input devices and interaction techniques can help to structure the interface such that there is a more direct match between the user's tasks and the low-level syntax of the individual actions that must be performed to achieve those tasks. The choice of device and technique thus directly influences the steps required of the user and hence the apparent complexity of an interface design (Buxton, 1986a).

EVALUATION AND ANALYSIS OF INPUT DEVICES

Beyond standard usability engineering techniques (see Part VI: The Development Process), a number of techniques tailored to the study of input devices also exist. Representative tasks (Buxton, 1995c), such as target acquisition, pursuit tracking, freehand drawing, and dragging versus tracking performance (MacKenzie et al., 1991), can be used to formally or informally evaluate devices. Here, we focus on formal analysis using Fitts' Law, the Steering Law, and the Keystroke-Level Model.

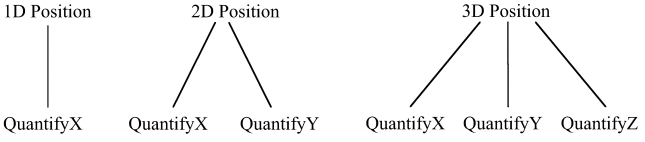


FIGURE 9.5. Task hierarchies for 1D, 2D, and 3D position tasks.

Fitts' Law and Hick's Law

Fitts' Law (Fitts, 1954) is an experimental paradigm that has been widely applied to the comparison and optimization of pointing devices. Fitts' Law is used to measure how effectively a pointing device can acquire targets on the screen. Fitts' Law was first applied to the study of input devices by Card, English, and Burr (1978); it is now a standard for device comparisons (Douglas, Kirkpatrick, & MacKenzie, 1999). Fitts' Law applies to a remarkably diverse task conditions, including (a) rate-controlled devices (MacKenzie, 1992a), (b) area cursors (Kabbash & Butxon, 1995), (c) scrolling (Hinckley et al., 2002), and (d) zooming (Guiard, Buourgeois, Mottet, & Beaudouin-Lafon, 2001). For further guidance on conducting Fitts' Law studies, see Douglas and colleagues (1999), MacKenzie (1992b), and Raskin (2000).

The standard Fitts' task paradigm measures the movement time *MT* between two targets separated by amplitude *A*, with a width *W* of error tolerance (Fig. 9.6).

Fitts' Law states that a logarithmic function of the ratio of *A* to *W* predicts the *average* movement time *MT*. The Fitts' Law formulation typically used for input device studies is

$$MT = a + b \log_2(A/W + 1) \quad (9.1)$$

Here, the constants *a* and *b* are coefficients fit to the average of all observed *MT* for each combination of *A* and *W* in the experiment. One calculates a and b via linear regression using a statistical package or spreadsheet. The constants *a* and *b* depend heavily on the exact task setting and input device, so be wary of substituting "typical" values for these constants, or of comparing constants derived from different studies.

Psychomotor interpretations for Fitts' Law have been proposed (Douglas & Mithal, 1997); however, since the law characterizes the central tendency of a large number of pointing movements, the law may simply reflect information-theoretic entropy (MacKenzie, 1989). For example, Hick's Law, a model of decision time for a set of choices (e.g., in a menu), has a general form almost identical to Fitts' Law:

$$H = \log_2(n + 1) \quad (9.2)$$

Here, *n* is the number of equally probably alternatives, and *H* is the entropy of the decision. If we view Fitts' task (0) as a "decision" along the amplitude *A* between *n* discrete targets of width *W*, this raises the possibility that Fitts' Law and Hick's Law are fundamentally the same law where $n = A/W$.

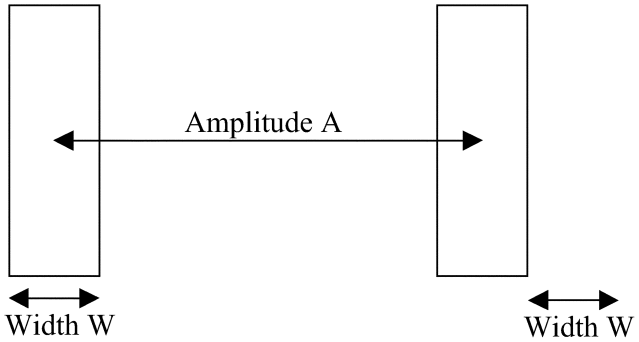


FIGURE 9.6. Fitts' task paradigm (Fitts, 1954).

The Steering Law and Minimum Jerk Law

Steering a cursor through a narrow tunnel, as required to navigate a pull-down menu, is not a Fitts' task because steering requires a continuous accuracy constraint: the cursor must stay within the tunnel at all times. For a straight-line tunnel (Fig. 9.7) of width *W* and length *A*, for example, the Steering Law predicts that movement time is a linear function of *A* and *W*:

$$MT = a + b\, A/W \quad (9.3)$$

The Steering Law can also model arbitrary curved paths, as well as instantaneous velocity (Accot & Zhai, 1997). A limitation of the Steering Law is that it only models successful completion of the task; errors are not considered.

The Minimum Jerk Law (Viviani & Flash, 1995) characterizes the dynamics of motions that may follow a curved path but do not have a continuous accuracy constraint. The law states that unconstrained human movement trajectories tend to minimize the derivative of acceleration (jerk); one of its implications is that there is a two-thirds power law linking tangential velocity and path curvature; however, no one has yet formulated a universal law that handles varying accuracy constraints and curvature (Lank & Saund, 2005).

The Keystroke-Level Model (KLM) and GOMS Analysis

The KLM is an engineering and analysis tool that can be used to estimate the time needed for experts to complete a routine task (Card, Moran, & Newell, 1980). To apply the KLM, count the elemental inputs required to complete a task, including (a) keystrokes, (b) homing times to acquire input devices, (c) pauses for mental preparation, and (d) pointing at targets. For each elemental input, substitute a constant estimate of the average time required using the values from Card et al. (1980), or by collecting empirical data (Hinckley et al., 2006), and sum them to yield an overall time estimate. The model assumes error-free execution, so it cannot estimate time for the problem-solving behaviors of novices, but it does employ several heuristics to model mental pauses (Raskin, 2000).

GOMS (Goals, Objects, Methods, and Selection rules) models extend the keystroke-level model (John & Kieras, 1996). Some GOMS models can account for user knowledge and in-

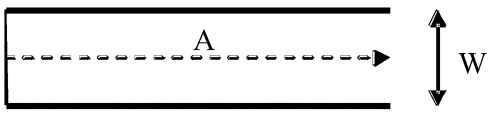


FIGURE 9.7. The Steering Law for a straight tunnel. The user must follow the dotted line without moving beyond the borders. Adapted from (Accot & Zhai, 1999).

terleaving of tasks, but are more difficult to apply than the KLM. Both GOMS and KLM models are engineering tools that produce *estimates* for expert completion time of routine tasks. These models do not replace the need for usability testing and evaluation, but they do offer a means to assess a design without implementing software, training end users, and evaluating their performance (Olson & Olson, 1990). Physical articulation times derived from KLM or GOMS analyses can also be used to help interpret results of empirical studies (Hinckley et al., 2006; see also chapter 60, "Model-based Evaluations"; and chapter 5, "Cognitive Architecture".

TRANSFER FUNCTIONS: HOW TO TRANSFORM AN INPUT SIGNAL

A transfer function is a mathematical transformation that scales the data from an input device. Typically, the goal is to provide more stable and more intuitive control, but one can easily design a poor transfer function that hinders performance. A transfer function that matches the properties of an input device is known as an *appropriate mapping*. For force sensing input devices, the transfer function should be a *force-to-velocity* function; for example, the force one exerts on the IBM Trackpoint isometric joystick controls the speed at which the cursor moves. Other appropriate mappings include *position-to-position* or *velocity-to-velocity* functions, used with tablets and mice, respectively.

A common example of an inappropriate mapping is calculating a velocity based on the position of the mouse cursor, such as to scroll a document. The resulting input is difficult to control, and this inappropriate rate mapping is only necessary because the operating system clips the cursor to the screen edge. A better solution would be to ignore the cursor position and instead use the relative position information reported by the mouse to directly control the change of position within the document.

Self-centering devices. Rate mappings suit force-sensing devices or other devices that return to center when released (Zhai, 1993; Zhai et al., 1997). This property allows the user to stop quickly by releasing the device. The formula for a nonlinear rate mapping is

$$dx = K x^{\alpha} \tag{9.4}$$

Where x is the input signal, dx is the resulting rate, K is a gain factor, and α is the nonlinear parameter. The best values for K and α depend on the details of the device and application, and appropriate values must be identified by experimentation or optimal search (Zhai & Milgram, 1993). Many commercial devices use more complex mappings (Rutledge & Selker, 1990).

Motion-sensing devices. Desktop systems use an exponential transformation of the mouse velocity, known as an *acceleration* function, to modify the cursor response (Microsoft Corp., 2002). Acceleration functions do not directly improve pointing performance, but do limit the footprint required by a device (Jellinek & Card, 1990), which may lead to greater comfort or less frequent clutching (Hinckley et al., 2002).

Absolute devices. It is possible to temporarily violate the 1:1 control-to-display mapping of absolute devices such as touchscreens by damping small motions to provide fine adjustments; large motions revert to an absolute 1:1 mapping (Sears & Shneiderman, 1991). A drawback of this technique is that cursor feedback in the tracking state becomes the default behavior of the device, rather than dragging (Buxton, 1990), but researchers are exploring ways to overcome this (Benko, Wilson, & Baudisch, 2006).

FEEDBACK: WHAT HAPPENS IN RESPONSE TO AN INPUT?

From the technology perspective, one can consider feedback as active or passive. Active feedback is under computer control; passive feedback is not, and may result from internal sensations within the user's own body, such as muscle tension from holding down a button, or physical properties of the device, such as the feel of clicking its buttons.

The industrial design suggests how to use a device before a user touches it (Norman, 1990). Mechanical sounds and vibrations produced by a device provide positive feedback for the user's actions (Lewis, Potosnak, & Magyar, 1997). The shape of the device and the presence of landmarks can help users acquire a device without having to look at it (Hinckley, Pausch, Proffitt, & Kassell, 1998b).

Proprioceptive and Kinesthetic Feedback

Internal sensations of body posture, motion, and muscle tension (Burdea, 1996; Gibson, 1962) may allow users to feel how they are moving an input device without either looking at the device or receiving visual feedback on a display. This is important when the user's attention is divided between multiple tasks and devices (Balakrishnan & Hinckley, 1999; Fitzmaurice & Buxton, 1997; Mine, Brooks, & Sequin, 1997). Muscular tension can help to phrase together multiple related inputs (Buxton, 1986a) and may make mode transitions more salient to the user (Hinckley et al., 2006; Raskin, 2000; Sellen, Kurtenbach, & Buxton, 1992).

Kinesthetic Correspondence

Graphical feedback on the screen should correspond to the direction that the user moves the input device (Britton, Lipscomb, & Pique, 1978). If the user moves a device to the left, then the object on the screen should likewise move left; however, users can easily adapt to certain kinds of noncorrespondences: when the user moves a mouse forward and backward, the cursor actually moves up and down on the screen; if the user drags a scrollbar downward, the text on the screen scrolls upwards. Researchers have also found that the dimensions of an input de-

vice should match the perceptual structure of a task (Jacob, Sibert, McFarlane, & Mullen, 1994).

Snapping Behaviors and Active Haptic Feedback

Software constraints, such as snapping (Baudisch, Cutrell, Hinckley, & Eversole, 2005), often suffice to support a user's tasks. Active force or tactile feedback (Burdea, 1996) can provide attractive forces for a target, or additional feedback for the boundaries of a target, but such feedback typically yields little or no performance advantage even for isolated target selection (Akamatsu & Mackenzie, 1996; MacKenzie, 1995). Such techniques must evaluate selection among multiple targets, because haptic feedback or snapping behavior for one target interferes with the selection of others (Grossman & Balakrishnan, 2005; Oakley, Brewster, & Gray, 2001). *Visual dominance* refers to the tendency for vision to dominate other modalities (Wickens, 1992); haptic feedback typically must closely match visual feedback, which limits its utility as an independent modality (Campbell, Zhai, May, & Maglio, 1999). One promising use of tactile feedback is to improve state transitions (Poupyrev & Maruyama, 2003; Snibbe & MacLean, 2001). For further discussion of active feedback modalities, see chapter 12, Haptic Interfaces, and chapter 13, "Nonspeech Auditory Output."

KEYBOARDS AND TEXT ENTRY

Typewriters have been in use for over 100 years; the QWERTY key layout dates to 1868 (Yamada, 1980). Despite the antiquity of the design, QWERTY keyboards are extremely well suited to human performance and are unlikely to be supplanted by new key layouts, speech recognition technologies, or other techniques any time soon. Many factors can influence typing performance, including the size, shape, activation force, key travel distance, and the tactile and auditory feedback provided by striking the keys (Lewis et al., 1997), but these well-established design details are not our focus here.

Procedural Memory

Procedural memory allows performance of complex sequences of practiced movements, seemingly without any cognitive effort (Anderson, 1980). Procedural memory enables touch typing on a keyboard with minimal attention while entering commonly used symbols. As a result, users can focus attention on mental composition and verification of the text appearing on the screen. Dedicated or chorded key presses for frequently used commands (hotkeys) likewise allow rapid command invocation (McLoone, Hinckley, & Cutrell, 2003). The automation of skills in procedural memory is described by the *power law of practice*:

$$T = aP^b \tag{9.5}$$

Here, T is the time to perform a task, P is the amount of practice, and the multiplier a and exponent b are fit to the observed data (Anderson, 1980). For a good example of applying the power law of practice to text entry research, see MacKenzie, Kober, Smith, Jones, and Skepner (2001).

Alternative keyboard layouts such as Dvorak offer about a 5% performance gain (Lewis et al., 1997), but the power law of practice suggests this small gain comes at a substantial cost for retraining time; however, ergonomic QWERTY keyboards do preserve much of a user's skill for typing. These split-angle keyboards are not faster, but some can help maintain neutral posture of the wrist and thus avoid ulnar deviation (Honan, Serina, Tal, & Rempel, 1995; Marklin, Simoneau, & Monroe, 1997; Smutz, Serina, Bloom, & Rempel, 1994), which has been associated with increased pressure in the carpal tunnel (Putz-Anderson, 1988; Rempel, Bach, Gordon, & Tal, 1998).

Mobile Text Entry, Character Recognition, and Handwriting Recognition

The difficulty of entering text on handheld devices and cell phones has led to many new text-entry techniques (MacKenzie, 2002), but most offer only 10–20 words-per-minute (wpm) typing rates, compared to approximately 60 wpm for a touch typist.

Many designs for cell phones and other handheld devices, such as the RIM Blackberry, offer two-thumb keyboards with QWERTY key layouts. The principal virtue of QWERTY is that common pairs of letters tend to occur on opposite hands. This alternation is a very efficient movement pattern for both standard and two-thumb keyboards, since one hand completes a key press while the other hand moves to the next key (MacKenzie & Soukoreff, 2002). A recent study found that two-thumb keyboards offer text entry rates approaching 60 wpm (Clarkson, Clawson, Lyons, & Starner, 2005). This suggests that one-handed text entry rates are fundamentally limited due to the serial nature of character entry, despite novel improvements (MacKenzie et al., 2001; Wigdor & Balakrishnan, 2003). Word prediction may help, but also requires overhead for users to monitor and decide whether to use the predictions.

Soft keyboards and character recognition techniques are popular for pen-operated devices, but likewise are limited to serial entry. Soft keyboards depict keys in a graphical user interface to allow typing with a touchscreen or stylus. Design issues for soft keyboards differ from mechanical keyboards. Soft keyboards demand visual attention because the user *must look at the keyboard* to aim the pointing device. Only one key at a time can be touched, so much of the time is spent moving back and forth between keys (Zhai, Hunter, & Smith, 2000). A soft keyboard can allow the user to draw gestures across multiple keys; in combination with a language model, this can allow entire words to be entered with a single pen gesture (Kristensson & Zhai, 2004).

Handwriting (even on paper, with no "recognition" involved) proceeds at about 15 wpm. Ink has significant value as a natural data type without recognition: it offers an expressive mix of writing, sketches, and diagrams. Although recognition technology continues to improve, recognizing natural handwriting remains difficult and error prone for computers, and demands error correction input from the user. To make performance more pre-

dictable for the user, some devices instead rely on single-stroke gestures, known as *unistrokes* (Goldberg & Richardson, 1993), including *graffiti* for the PalmPilot. Unistroke alphabets attempt to strike a design balance such that each letter is easy for a computer to distinguish, yet also straightforward for users to learn (MacKenzie & Zhang, 1997).

MODALITIES OF INTERACTION

In the search for designs that enhance interfaces and enable new usage scenarios, researchers have explored several strategies that transcend any specific type of device.

Speech and Multimodal Input

Speech has substantial value without recognition. Computers can augment human-human communication across both time and space by allowing users to record, edit, replay, or transmit digitized speech and sounds (Arons, 1993; Buxton, 1995b; Stifelman, 1996). Systems can also use microphone input to detect ambient speech and employ this as a cue to help prioritize notifications (Horvitz, Jacobs, & Hovel, 1999; Sawhney & Schmandt, 2000; Schmandt, Marmasse, Marti, Sawhney, & Wheeler, 2000).

Computer understanding of human speech does not enable users to talk to a computer as one would converse with another person (but see also chapter 19, Conversational Interfaces and Technologies). Speech recognition can succeed for a limited vocabulary, such as speaking the name of a person from one's contact list to place a cell phone call; however, error rates increase as the vocabulary and complexity of the grammar grows, if the microphone input is poor, or if users employ "out-of-vocabulary" words. It is difficult to use speech to refer to spatial locations, so it cannot eliminate the need for pointing (Cohen & Sullivan, 1989; Oviatt, DeAngeli, & Kuhn, 1997; see also chapter 21, "Multimodal Interfaces"). Currently, keyboard-mouse text entry for the English language is about twice as fast as automatic speech recognition (Karat, Halverson, Horn, & Karat, 1999); furthermore, speaking can interfere with one's ability to compose text and remember words (Karl, Pettey, & Shneiderman, 1993). Finally, speech is inherently nonprivate in public situations. Thus, speech has an important role to play, but claims that speech will soon supplant manual input devices should be considered with skepticism.

Bimanual Input

People use both hands to accomplish most real-world tasks (Guiard, 1987), but computer interfaces make little use of the nonpreferred hand for tasks other than typing. Bimanual input enables compound input tasks such as navigation/selection tasks, where the user can scroll with the nonpreferred hand while using the mouse in the preferred hand (Buxton & Myers, 1986). This assignment of roles to the hands corresponds to Guiard's kinematic chain theory (Guiard, 1987): the nonpreferred hand sets a frame of reference (scrolling to a location in the document) for the action of the preferred hand (selecting an item within the page using the mouse). Other applications for bimanual input include command selection (Bier, Stone, Pier, Buxton, & DeRose, 1993; Kabbash, Buxton, & Sellen, 1994), drawing tools (Kurtenbach, Fitzmaurice, Baudel, & Buxton, 1997), and virtual camera control and manipulation (Balakrishnan & Kurtenbach, 1999; Hinckley et al., 1998b). Integrating additional buttons and controls with keyboards to encourage bimanual interaction can also improve the efficiency of some common tasks (MacKenzie & Guiard, 2001; McLoone et al., 2003).

Pen and Gesture Input

The Palm Pilot and Tablet PC have led to a renaissance in pen and gesture research. Pens lend themselves to command gestures analogous to proofreader's marks, such as crossing out a word to delete it. Note that in this example, the gesture integrates the selection of a *delete* command with the selection of the word to be deleted. Another example is moving a paragraph by circling it and drawing a line to its new location. This integrates the verb, object, and indirect object by specifying the command, the extent of text to move, and the new location for the text (Hinckley et al., 2005a; Kurtenbach & Buxton, 1991). *Marking menus* use straight-line gestures along the primary compass directions for rapid command selection (Kurtenbach, Sellen, & Buxton, 1993; Zhao & Balakrishnan, 2004).

Pen interfaces must decide whether to treat pen strokes as *ink* content or as *gesture* commands. Some applications avoid this recognition problem by treating all strokes as commands (Kurtenbach & Buxton, 1991), but for a free-form drawing or note-taking application, users need to interleave ink content and command input. The status-quo solution presents commands in a toolbar or menu at the edge of the screen; however, this necessitates round trips between the work area and the command area (Fitzmaurice, Khan, Pieke, Buxton, & Kurtenbach, 2003a), which becomes inconvenient in direct proportion to the display size. Pressing a button with the nonpreferred hand is a fast and robust means to switch between ink and gesture modes (Li et al., 2005).

Techniques to automatically distinguish ink and gestures have been proposed, but only for highly restricted gesture sets (Saund & Lank, 2003). Punctuation (tapping) has also been explored as a way to both identify and delimit command phrases (LaViola, 2004). A fundamental problem with both of these approaches is that the system cannot classify a set of strokes as a gesture or as ink until after the user has finished drawing the entire command phrase. This makes it difficult to provide interactive feedback or to prompt the user with the available commands before the user commits to an operation.

While moving the pen to toolbars at the edge of the screen seems slow on a tablet computer, in practice, this "round trip strategy" (Fitzmaurice et al., 2003a) is difficult to improve upon. On a tablet the size of a standard 8.5 × 11 inch sheet of paper, a round trip requires approximately 1.5 seconds; however, the user can mentally prepare for the next step of the interaction while moving the pen. A locally drawn gesture (such as a straight-line marking menu command) may take less time to articulate,

but thinking about what command to select requires additional time unless the task is a routine one. Pressing a button for gesture mode also requires some overhead, as does lifting the pen at the end of the gesture. Also note that performing a sequence of gestures (e.g., tagging words in a document as keywords by circling them) requires time to travel between screen locations. The round-trip strategy absorbs this travel time into the round trip itself, but with gestures, this is an extra cost that reduces the benefit of keeping the interaction localized.

Thus, on a tablet-sized device, it is difficult to realize a substantial time savings just by reducing round trips. For our hypothetical task of tagging keywords in a document, Fig. 9.8 illustrates this predicament for average task times drawn from recent studies (Hinckley et al., 2006; Li et al., 2005). The chart shows two successive command selections, and assumes some mental preparation is required before issuing each command. Thus, the potential benefit of pen gestures depends on the sequence of operations as well as the elimination of *multiple* round trips, as may be possible with techniques that integrate selection of verb, object, and indirect object (Hinckley et al., 2005a; Kurtenbach & Buxton, 1991). Localized interaction may also offer indirect benefits by reducing physical effort and by keeping the user's visual attention focused on their work (Grossman, Hinckley, Baudisch, Agrawala, & Balakrishnan, 2006; Kabbash et al., 1994).

Whole Hand Input

Humans naturally gesture and point using their hands during verbal communication, which has motivated research into free-hand gestures, often in combination with speech recognition (Bolt, 1980; Hauptmann, 1989; Wilson & Oliver, 2003). Cadoz categorized hand gestures as semiotic, ergotic, or epistemic. Semiotic gestures, such as "thumbs up," communicate information (Rime & Schiaratura, 1991). Ergotic gestures manipulate physical objects. Epistemic gestures are exploratory motions that gather information (Kirsh, 1995; Kirsh & Maglio, 1994). The interaction literature focuses on empty-handed semiotic gestures (Freeman & Weissman, 1995; Jojic, Brumitt, Meyers, & Harris, 2000; Maes, Darrell, Blumberg, & Pentland, 1997). A major challenge is to correctly identify when a gesture, as opposed to an incidental hand movement, starts and stops (Baudel & Beaudouin-Lafon, 1993; Wilson & Oliver, 2003). The lack of deterministic state transitions (Buxton, 1990; Vogel & Balakrishnan, 2005) can lead to errors of user intent or errors of computer interpretation (Bellotti, Back, Edwards, Grinter, Lopes, & Henderson, 2002). Other problems include fatigue from extending one's arms for long periods, and the imprecision of pointing at a distance. By contrast, tangible interaction techniques (Ishii & Ullmer, 1997) and augmented devices (Harrison, Fishkin, Gujar, Mochon, & Want, 1998) sense ergotic gestures via a physical intermediary (Hinckley et al., 1998b; Zhai et al., 1996). The emergence of cameras, cell phones, and tablets augmented with accelerometers and other sensors suggest this is a promising design space.

Background Sensing Techniques

Sensors can enable a mobile device to sense when the user picks up, puts down, looks at, holds, or walks around with the device. These actions give a device information about the context of its use, and represent a hidden vocabulary of naturally oc-

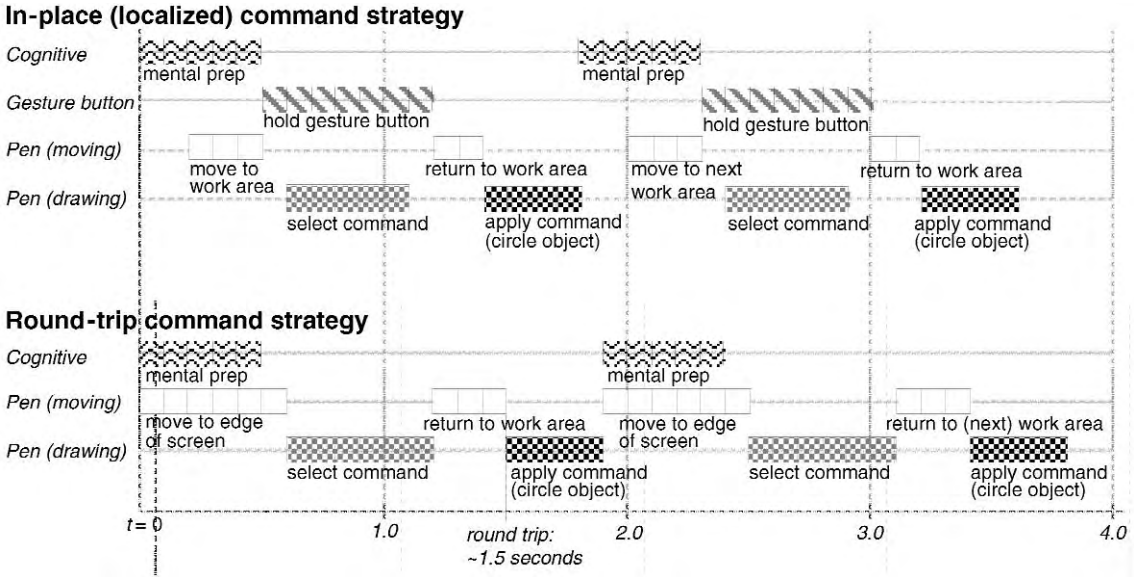


FIGURE 9.8. Chart comparing task times for performing two successive commands on a tablet, with either in-place command selection (top) or round-trip command selection (bottom). The individual boxes in each subtask represent 100 millisecond intervals.

curring gestures that people spontaneously exhibit in day-to-day activity. For example, commercially available digital cameras now employ a tilt sensor to detect the orientation of the camera, and use this to automatically save photos in the correct orientation, as well as to interactively switch the display between portrait/landscape formats (Hinckley et al., 2005b; Hinckley, Pierce, Sinclair, & Horvitz, 2000). Here the sensor allows the device to adapt its behavior to the user's needs, rather than requiring the user to take extra steps to control the photo orientation and display format (Buxton, 1995a).

Sensors can also be embedded in the environment. When one walks into a modern department store, there is no explicit command to open the doors: the doors sense motion and automatically open. Researchers are investigating new ways to leverage such contextual sensing to enrich and simplify interaction with devices and digital environments (Abowd & Mynatt, 2000; Schilit, Adams, & Want, 1994).

Multitouch Tables and Screens

Technical advances have led to much recent research in touch-sensitive tables and projection screens. These technologies blur the distinction between whole-hand gestural input and traditional single-point touchscreens and touch tablets. Recent prototype systems demonstrate capture of the shape formed by the hand(s) in contact with a surface (Wilson, 2005), multiple points of contact (Han, 2005), or even images of objects placed on or near a surface (Matsushita & Rekimoto, 1997; Wilson, 2004). The DiamondTouch (Dietz & Leigh, 2001) is unique in its ability to determine which user produces each point of contact, which has led to a number of innovative applications and techniques (Shen, Everitt, & Ryall, 2003; Wu & Balakrishnan, 2003). For an overview of design issues for tabletop systems, see Scott, Grant, and Mandryk (2003).

A Society of Devices

Wireless networking is the technology that will most disrupt traditional approaches to human-computer interaction in the coming years, because it breaks down barriers between devices and enables a new society of devices that fill specific roles, yet can still coordinate their activities (Fitzmaurice, Khan, Buxton, Kurtenbach, & Balakrishnan, 2003b; Want & Borriello, 2000). The interaction designer thus must consider the full range of scale for display sizes and form-factors that may embody an interaction task, as well as the interactions between different types of devices. How can interaction migrate from watches, cell phones, handheld devices, and tablets, all the way up to desktop monitors, digital whiteboards, and interactive wall-sized displays?

As digital devices become physically smaller, the displays and input mechanisms they offer are shrinking. Considerable effort has been devoted to supporting web browsing in limited screen space (Buyukkokten, Garcia-Molina, & Paepcke, 2001; Jones, Marsden, Mohd-Nasir, Boone, & Buchanan, 1999; Trevor, Hilbert, Schilit, & Koh, 2001). Techniques to make small displays virtually larger include (a) peephole displays (Fitzmaurice, 1993; Yee, 2003), (b) transparent overlays (Harrison, Ishii, Vicente, & Buxton, 1995; Kamba, Elson, Harpold, Stamper, & Sukaviriya, 1996), and (c) using on-screen visuals to suggest the locations of off-screen objects (Baudisch & Rosenholtz, 2003). Touchscreens and touch-sensitive controls minimize the vertical profile of devices (Wherry, 2003), but may suffer from inadvertent activation. Physical manipulations such as tilting that use the device itself as an interface seem particularly well suited to small devices (Harrison et al., 1998; Hinckley et al., 2000; Rekimoto, 1996). Tiny, bright, and inexpensive laser or LED projectors are just around the corner; progress on computer vision techniques suggests that interactive projection may allow small devices to project large displays and sensing surfaces (Raskar et al., 2004; Wilson, 2005).

At the other end of the spectrum, large-format displays are now affordable and common. Large displays lend themselves to collaboration and sharing of information with groups (Funkhouser & Li, 2000; Swaminathan & Sato, 1997), as well as giving a substantial physical presence to virtual activities (Buxton, Fitzmaurice, Balakrishnan, & Kurtenbach, 2000; Trimble, Wales, & Gossweiler, 2003). Researchers have explored pen and touch-screen interaction techniques for large displays (Guimbretiere, Stone, & Winograd, 2001; Moran et al., 1997). Unfortunately, many technologies sense only a single point of contact. For interaction at a distance with large displays, it remains unclear what interaction techniques work best (Olsen & Nielsen, 2001; Vogel & Balakrishnan, 2004; Vogel & Balakrishnan, 2005); even when a user is close to a large display, interacting with portions of the display that are out view or beyond arm's length raises challenges (Bezerianos & Balakrishnan, 2005; Khan, Matejka, Fitzmaurice, & Kurtenbach, 2005).

Displays of various sizes support different activities and social conventions; one of the principle challenges of ubiquitous computing (Weiser, 1991) is finding techniques that make it easy for users to work within a digital ecology that supports a range of tasks spanning multiple computers, displays, and interactive surfaces. Even on a single computer, users do not treat multiple monitors as one large display, but instead employ the boundary between displays to partition their tasks (Grudin, 2001). Several projects have probed how to use small displays as adjuncts to large ones (Myers, Lie, & Yang, 2000; Myers, Stiel, & Garguilo, 1998; Rekimoto, 1998), allowing simultaneous interaction with private information on a personal device, and a shared or public context on a larger display. Users need techniques that allow them to access and share information across the boundaries of individual devices, as well as to dynamically bind together multiple devices and displays to accomplish their tasks (Hinckley, Ramos, Guimbretiere, Baudisch, & Smith, 2004; Rekimoto, Ayatsuka, & Kohno, 2003). Such interaction techniques inherently involve multiple persons, and thus must consider how people use physical proximity and relative body orientation, a field known as *proxemics* (Deasy & Lasswell, 1985; Hinckley et al., 2004; Sommer, 1965).

CURRENT AND FUTURE TRENDS FOR INPUT

The designer of an interactive system should take a broad view of input, and consider not only traditional pointing techniques and graphical user interface widgets, but also issues such as

search strategies to access information in the first place, sensor inputs that enable entirely new data types, synthesis techniques to extract meaningful structure from data, and integration of traditional technologies such as paper that offer fundamental strengths.

Good search tools may reduce the many inputs needed to manually search and navigate file systems. Knowledge work requires integration of external information from web pages or databases (Yee, Swearingen, Li, & Hearst, 2003) as well as reuse of personal information from documents, electronic mail messages, and other content authored or viewed by a user (Dumais, Cutrell, Cadiz, Jancke, Sarin, & Robbins, 2003; Lansdale & Edmonds, 1992). Unified full-text indexing allows users to quickly query their personal information across multiple information silos, and can present information in the context of memory landmarks such as the date a message was sent, who authored the text, or the application used to create a document (Cutrell, Robbins, Dumais, & Sarin, 2006).

New sensor inputs such as location and tagging technologies are coming to fruition. Radio-frequency identification (RFID) tags (Want, Fishkin, Gujar, & Harrison, 1999) allow computers to identify tagged physical objects, thus enabling manipulation of ordinary objects as an input for computers. A mobile tag reader can identify tagged physical locations (Smith, Davenport, & Hwa, 2003). Wireless communication technologies are poised to deliver ubiquitous location-based services (Schilit et al., 1994; Want, Hopper, Falcao, & Gibbons, 1992). Cell phones and low-power radios for wireless networking can sense their location or proximity to other devices via analysis of signal strengths (Bahl & Padmanabhan, 2000; Krumm & Hinckley, 2004). As another example, attempting to type a secure password on a mobile phone keypad quickly convinces one that biometric sensors or some other convenient means for establishing identity is essential for these devices to succeed. Such sensors could also make services such as personalization of interfaces much simpler.

The need to extract models and synthesize structure from large quantities of low-level inputs suggests that data mining and machine learning techniques will become important adjuncts to interaction (Fails & Olsen, 2003; Fitzmaurice, Balakrisnan, & Kurtenbach, 1999; Horvitz, Breese, Heckerman, Hovel, & Rommelse, 1998). Whenever a system considers automatic actions on behalf of the user, however, an important design principle in the face of uncertainty is to "do less, but do it well" (Horvitz, 1999).

With so many new technologies, it is easy to forget that paper remains an important input and display medium (Sellen & Harper, 2002). Paper is inexpensive, easy to annotate, rapid to access, comfortable to read, light to carry, and can be accessed after tens or even hundreds of years. Because technology will not replace paper any time soon, researchers are studying how to bring paperlike interfaces to digital devices (Schilit, Golovchinsky, & Price, 1998; Wolf, Rhyne, & Ellozy, 1989), as well as how to augment paper with digital capabilities (Guimbretiere, 2003; Liao, Guimbretiere, & Hinckley, 2005; Stifelman, 1996).

We must make substantial progress in all of these areas to advance human interaction with technology. The forms and capabilities of these and other technologies will continue to advance, but human senses and cognitive skills will not. We will continue to interact with computers using our hands and physical intermediaries, not necessarily because our technology requires us to do so, but because touching, holding, and moving physical objects is the foundation of the long evolution of tool use in the human species (Wilson, 1998).

References

Abowd, G., & Mynatt, E. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction, 7*(1), 29–58.

Accot, J., & Zhai, S. (1997). Beyond Fitts' Law: Models for trajectory-based HCI tasks. *Proceedings of the CHI '97 ACM Conference on Human Factors in Computing Systems* (pp. 295–302).

Accot, J., & Zhai, S. (1999). Performance evaluation of input devices in trajectory-based tasks: An application of the Steering Law. *Proceedings of the CHI '99* (pp. 466–472).

Accot, J., & Zhai, S. (2001). Scale effects in Steering Law tasks. *Proceedings of CHI 2001 ACM Conference on Human Factors in Computing Systems*, (pp. 1–8).

Akamatsu, M., & Mackenzie, I. S. (1996). Movement characteristics using a mouse with tactile and force feedback. *International Journal of Human-Computer Studies, 45*, 483–493.

Anderson, J. R. (1980). Cognitive skills. In *Cognitive psychology and its implications* (pp. 222–254). San Francisco: W. H. Freeman.

Apitz, G., & Guimbretiere, F. (2004). CrossY: A crossing based drawing application. *UIST 2004*, 3–12.

Arons, B. (1993). SpeechSkimmer: Interactively skimming recorded speech. In *UIST'93 Symposium on User Interface Software & Technology* (pp. 187–195).

Bahl, P., & Padmanabhan, V. (2000). RADAR: An in-building RF-based user location and tracking system. *Proceedings of the IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, 775–784.

Balakrishnan, R., Baudel, T., Kurtenbach, G., & Fitzmaurice, G. (1997). The Rockin'Mouse: Integral 3D manipulation on a plane. *Proceedings of the CHI '97 Conference on Human Factors in Computing Systems*, 311–318.

Balakrishnan, R., & Hinckley, K. (1999). The role of kinesthetic reference frames in two-handed input performance. *Proceedings of the ACM UIST'99 Symposium on User Interface Software and Technology*, 171–178.

Balakrishnan, R., & Kurtenbach, G. (1999). Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *Proceedings of the CHI'99 ACM Conference on Human Factors in Computing Systems*, 56–63.

Baudel, T., & Beaudouin-Lafon, M. (1993). Charade: Remote control of objects using hand gestures. *Communications of the ACM, 36*(7), 28–35.

Baudisch, R., Cutrell, E., Hinckley, K., & Eversole, A. (2005). Snap-and-go: Helping users align objects without the modality of traditional snapping. *Proceedings of the CHI 2005*, 301–310.

Baudisch, P., & Rosenholtz, R. (2003). Halo: a technique for visualizing off-screen objects. *Proceedings of the CHI '03*, 481–488.

Bellotti, V., Back, M., Edwards, W. K., Grinter, R., Lopes, C., & Henderson, A. (2002). Making sense of sensing systems: Five questions for

designers and researchers. *Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems*, 415–422.

Benko, H., Wilson, A., & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. *Proceedings of the CHI 2006*.

Bezerianos, A., & Balakrishnan, R. (2005). The vacuum: Facilitating the manipulation of distant objects. *Proceedings of the CHI '05*, 361–370.

Bier, E., Stone, M., Pier, K., Buxton, W., & DeRose, T. (1993). Toolglass and magic lenses: The see-through interface. *Proceedings of SIGGRAPH 93*, 73–80.

Bolt, R. (1980, August). Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics*, 262–270.

Bowman, D., Kruijff, E., LaViola, J., & Poupyrev, I. (2004). 3D user interfaces: Theory and practice, Addison-Wesley, Boston.

Britton, E., Lipscomb, J., & Pique, M. (1978). Making nested rotations convenient for the user. *Computer Graphics*, *12*(3), 222–227.

Burdea, G. (1996). *Force and Touch Feedback for Virtual Reality*. New York: John Wiley & Sons.

Buxton, W. (1983). Lexical and pragmatic considerations of input structure. *Computer Graphics, 17*(1), 31–37.

Buxton, W. (1986a). Chunking and phrasing and the design of human-computer dialogues. *Proceedings of the IFIP 10th World Computer Congress*, 475–480.

Buxton, W. (1986b). There's more to interaction than meets the eye. In D. Norman and S. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction* (pp. 319–337). Hillsdale, NJ.

Buxton, W. (1990). A three-state model of graphical input. *Proceedings of the INTERACT'90*, 449–456.

Buxton, W. (1995a). Integrating the periphery and context: A new taxonomy of telematics. *Proceedings of Graphics Interface '95*, 239–246.

Buxton, W. (1995b). Speech, language and audition. In R. Baecker, J. Grudin, W. Buxton, & S. Greenberg (Eds.), *Readings in human-computer interaction: Toward the year 2000* (pp. 525–537). Morgan Kaufmann Publishers, San Francisco.

Buxton, W. (1995c). Touch, gesture, and marking. In R. Baecker, J. Grudin, W. Buxton, & S. Greenberg (Eds.), *Readings in human-computer interaction: Toward the year 2000* (pp. 469–482). Morgan Kaufmann Publishers, San Francisco.

Buxton, W., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2000, July–August). Large displays in automotive design. *IEEE Computer Graphics and Applications*, 68–75.

Buxton, E., Hill, R., & Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. *Computer Graphics, 19*(3), 215–224.

Buxton, W., & Myers, B. (1986). A study in two-handed input. *Proceedings of CHI '86: ACM Conference on Human Factors in Computing Systems*, 321–326.

Buyukkokten, O., Garcia-Molina, H., & Paepcke, A. (2001). Accordion summarization for end-game browsing on PDAs and cellular phones. *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems*.

Campbell, C., Zhai, S., May, K., & Maglio, P. (1999). What you feel must be what you see: Adding tactile feedback to the trackpoint. *Proceedings of INTERACT '99: 7th IFIP conference on Human Computer Interaction*, 383–390.

Card, S., English, W., & Burr, B. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, *21*, 601–613.

Card, S., Mackinlay, J., & Robertson, G. (1990). The design space of input devices. *Proceedings of ACM CHI '90 Conference on Human Factors in Computing Systems*, 117–124.

Card, S., Mackinlay, J., & Robertson, G. (1991). A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems, 9*(2), 99–122.

Card, S., Moran, T., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396–410.

Clarkson, E., Clawson, J., Lyons, K., & Starner, T. (2005). An empirical study of typing rates on mini-QWERTY keyboards. *CHI Extended Abstracts*, 1288–1291.

Cohen, P. R., & Sullivan, J. W. (1989). Synergistic use of direct manipulation and natural language. *Proceedings of the ACM CHI '89 Conference on Human Factors in Computing Systems*, 227–233.

Cutrell, E., Robbins, D., Dumais, S., & Sarin, R. (2006). Fast, flexible filtering with phlat: Personal search and organization made easy. *Proceedings of the CHI 2006*.

Deasy, C. M., & Lasswell, T. E. (1985). *Designing places for people: A handbook on human behavior for architects, designers, and facility managers*. New York: Whitney Library of Design.

Dietz, P., & Leigh, D. (2001). DiamondTouch: A multi-user touch technology. *CHI Letters, 3*(2), 219–226.

Dillon, R. F., Eday, J. D., & Tombaugh, J. W. (1990). Measuring the true cost of command selection: Techniques and results. *Proceedings of the ACM CHI '90 Conference on Human Factors in Computing Systems*, 19–25.

Douglas, S., Kirkpatrick, A., & MacKenzie, I. S. (1999). Testing pointing device performance and user assessment with the ISO 9241, Part 9 Standard. *Proceedings of the ACM CHI '99 Conference on Human Factors in Computing Systems*, 215–222.

Douglas, S., & Mithal, A. (1994). The effect of reducing homing time on the speed of a finger-controlled isometric pointing device. *Proceedings of the ACM CHI '94 Conference on Human Factors in Computing Systems*, 411–416.

Douglas, S. A., & Mithal, A. K. (1997). *Ergonomics of Computer Pointing Devices*, Springer-Verlag.

Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., & Robbins, D. (2003). Stuff I've seen: A system for personal information retrieval and re-use. *SIGIR 2003*, 72–79.

English, W. K., Englebart, D. C., & Berman, M. L. (1967). Display-selection techniques for text manipulation. *Transactions on Human Factors in Electronics, 8*(1), 5–15.

Fails, J. A., & Olsen, D. R. (2003). Interactive machine learning. *Proceedings from the ACM Intelligent User Interfaces (IUI '03)*, 39–45.

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*, 381–391.

Fitzmaurice, G., & Buxton, W. (1997). An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input. *Proceedings of CHI '97: ACM Conference on Human Factors in Computing Systems*, 43–50.

Fitzmaurice, G., Khan, A., Pieke, R., Buxton, B., & Kurtenbach, G. (2003a). Tracking menus. *UIST 2003*, 71–79.

Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM, 36*(7), 39–49.

Fitzmaurice, G. W., Balakrisnan, R., & Kurtenbach, G. (1999). Sampling, synthesis, and input devices. *Communications of the ACM, 42*(8) 54–63.

Fitzmaurice, G. W., Khan, A., Buxton, W., Kurtenbach, G., & Balakrishnan, R. (2003, November). Sentient Data. *Queue, 1*(8), 52–62.

Foley, J. D., Wallace, V., & Chan, P. (1984, November). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 13–48.

Freeman, W. T., & Weissman, C. (1995). Television control by hand gestures. *International Workshop on Automatic Face and Gesture Recognition*, 179–183.

Funkhouser, T., & Li, K. (2000). Large Format Displays. *IEEE Computer Graphics and Applications.* [Special Issue], (July–August), 20–75.

Gibson, J. (1962). Observations on active touch. *Psychological Review, 69*(6), 477–491.

Goldberg, D., & Richardson, C. (1993). Touch-typing with a stylus. *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, 80–87.

Grossman, T., & Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI '05*, 281–290.

Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., & Balakrishnan, R. (2006). Hover widgets: Using the tracking state to extend the capabilities of pen-operated devices. *CHI 2006*.

Grudin, J. (2001). Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. *CHI 2001*, 458–465.

Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *The Journal of Motor Behavior, 19*(4), 486–517.

Guiard, Y., Buourgeois, F., Mottet, D., & Beaudouin-Lafon, M. (2001). Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. *IHM-HCI 2001*.

Guimbretiere, F. (2003). Paper augmented digital documents. *UIST 2003*.

Guimbretiere, F., Stone, M. C., & Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. *Proceedings of the UIST 2001 Symposium on User Interface Software and Technology*, 21–30.

Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. *UIST 2005*, 115–118.

Harrison, B., Fishkin, K., Gujar, A., Mochon, C., & Want, R. (1998). Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *Proceedings of the ACM CHI '98 Conference on human factors in computing systems*, 17–24.

Harrison, B., Ishii, H., Vicente, K., & Buxton, W. (1995). Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. *Proceedings of the CHI '95 ACM Conference on Human Factors in Computing Systems*, 317–324.

Hauptmann, A. (1989). Speech and gestures for graphic image manipulation. *Proceedings of the CHI '89 ACM Conference on Human Factors in Computing Systems*, 241–245.

Hinckley, K., Baudisch, P., Ramos, G., & Guimbretiere, F. (2005a). Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. *Proceedings from the CHI 2005*, 451–460.

Hinckley, K., Cutrell, E., Bathiche, S., & Muss, T. (2002). Quantitative analysis of scrolling techniques. *CHI 2002.*

Hinckley, K., Czerwinski, M., & Sinclair, M. (1998a). Interaction and modeling techniques for desktop two-handed input. *Proceedings of the ACM UIST '98 Symposium on User Interface Software and Technology*, 49–58.

Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., & Cutrell, E. (2006). The springboard: Multiple modes in one spring-loaded control. *CHI 2006*.

Hinckley, K., Pausch, R., Goble, J. C., & Kassell, N. F. (1994). A survey of design issues in spatial input. *Proceedings of the ACM UIST '94 Symposium on User Interface Software and Technology*, 213–222.

Hinckley, K., Pausch, R., Proffitt, D., & Kassell, N. (1998b). Two-Handed Virtual Manipulation. *ACM Transactions on Computer-Human Interaction, 5*(3), 260–302.

Hinckley, K., Pierce, J., Horvitz, E., & Sinclair, M. (2005b). Foreground and background interaction with sensor-enhanced mobile devices [Special Issue]. *ACM TOCHI, 12*(1), 31–52.

Hinckley, K., Pierce, J., Sinclair, M., & Horvitz, E. (2000). Sensing techniques for mobile interaction. *ACM UIST 2000 Symposium on User Interface Software & Technology*, 91–100.

Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., & Smith, M. (2004). Stitching: Pen gestures that span multiple displays. *ACM 7th International Working Conference on Advanced Visual Interfaces* (*AVI 2004*), 23–31.

Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., & Conway, M. (1999). The VideoMouse: A camera-Based multi-degree-of-freedom input device. *ACM UIST '99 Symposium on User Interface Software & Technology*, 103–112.

Honan, M., Serina, E., Tal, R., & Rempel, D. (1995). Wrist postures while typing on a standard and split keyboard. *Proceedings of the HFES Human Factors and Ergonomics Society 39th Annual Meeting*, 366–368.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the ACM CHI '99 Conference on Human Factors in Computing Systems*, 159–166.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998, July). The Lumiere Project: Bayesian user modeling for inferring the goals and needs of software users. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 256–265). San Francisco: Morgan Kaufmann.

Horvitz, E., Jacobs, A., & Hovel, D. (1999). Attention-sensitive alerting. *Proceedings of the UAI '99 Conference on Uncertainty and Artificial Intelligence*, 305–313.

Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits, and atoms. *Proceedings of the CHI'97 ACM Conference on Human Factors in Computing Systems*, 234–241.

Jacob, R., Sibert, L., McFarlane, D., & Mullen, Jr., M. (1994). Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction, 1*(1), 3–26.

Jellinek, H., & Card, S. (1990). Powermice and user performance. *Proceedings of the ACM CHI '90 Conference on Human Factors in Computing Systems*, 213–220.

John, B. E., & Kieras, D. (1996). Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction*, *3*(4), 287–319.

Jojic, N., Brumitt, B., Meyers, B., & Harris, S. (2000). Detecting and estimating pointing gestures in dense disparity maps. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition.*

Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K., & Buchanan, G. (1999). Improving web interaction on small displays. *Computer Networks, 3*(11–16), 1129–1137.

Kabbash, P., & Butxon, W. (1995). The "Prince" Technique: Fitts' Law and selection using area cursors. *Proceedings of ACM CHI '95 Conference on Human Factors in Computing Systems*, 273–279.

Kabbash, P., Buxton, W., & Sellen, A. (1994). Two-handed input in a compound task. *Proceedings of the CHI '94 ACM Conference on Human Factors in Computing Systems*, 417–423.

Kamba, T., Elson, S. A., Harpold, T., Stamper, T., & Sukaviriya, P. (1996). Using small screen space more efficiently. *Proceedings of the Conference on Human Factors in Computing Systems*, 383.

Karat, C., Halverson, C., Horn, D., & Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. *Proceedings of the ACM CHI '99 Conference on Human Factors in Computing Systems*, 568–575.

Karl, L., Pettey, M., & Shneiderman, B. (1993). Speech-activated versus mouse-activated commands for word processing applications: An empirical evaluation. *International Journal of Man-Machine Studies, 39*(4), 667–687.

Khan, A., Matejka, J., Fitzmaurice, G., & Kurtenbach, G. (2005). Spotlight: Directing users' attention on large displays. *Proceedings from the CHI '05*, 791–798.

Kirsh, D. (1995). Complementary strategies: Why we use our hands when we think. *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, 212–217.

Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science, 18*(4), 513–549.

Kristensson, P., & Zhai, S. (2004). SHARK2: A large vocabulary shorthand writing system for pen-based computers. *Proceedings of the UIST 2004 Symposium on User Interface Software and Technology*, 43–52.

Krumm, J., & Hinckley, K. (2004). The NearMe Wireless Proximity Server. *Proceedings from the Ubicomp 2004.*

Kurtenbach, G., & Buxton, W. (1991). Issues in combining marking and direct manipulation techniques. *Proceedings from the UIST '91*, 137–144.

Kurtenbach, G., Fitzmaurice, G., Baudel, T., & Buxton, B. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *Proceedings of the CHI '97 ACM Conference on Human Factors in Computing Systems*, 35–42.

Kurtenbach, G., Sellen, A., & Buxton, W. (1993). An empirical evaluation of some articulatory and cognitive aspects of 'marking menus'. *J. Human Computer Interaction, 8*(1), 1–23.

Lank, E., & Saund, E. (2005). Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers and Graphics, 29*(4), 490–500.

Lansdale, M., & Edmonds, E. (1992). Using memory for events in the design of personal filing systems. *International Journal of Man-Machine Studies, 36*(1), 97–126.

LaViola, R. (2004). MathPad2: A system for the creation and exploration of mathematical sketches. *ACM Transactions on Graphics, 23*(3), 432–440.

Lewis, J., Potosnak, K., & Magyar, R. (1997). Keys and keyboards. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 1285–1316). Amsterdam: North-Holland.

Li, Y., Hinckley, K., Guan, Z., & Landay, J. A. (2005). Experimental analysis of mode switching techniques in pen-based user interfaces. *Proceedings from the CHI 2005*, 461–470.

Liao, C., Guimbretiere, F., & Hinckley, K. (2005). PapierCraft: A command system for interactive paper. *UIST 2005*, 241–244.

Lipscomb, J., & Pique, M. (1993). Analog input device physical characteristics. *SIGCHI Bulletin, 25*(3), 40–45.

MacKenzie, I. S. (1989). A note on the information-theoretic basis for Fitts' Law. *Journal of Motor Behavior, 21*, 323–330.

MacKenzie, I. S. (1992a). Fitts' Law as a research and design tool in human-computer interaction. *Human-Computer Interaction, 7*, 91–139.

MacKenzie, I. S. (1992b). Movement time prediction in human-computer interfaces. *Proceedings from the Graphics Interface '92*, 140–150.

MacKenzie, I. S. (1995). Input devices and interaction techniques for advanced computing. In W. Barfield, & T. Furness (Eds.), *Virtual environments and advanced interface design* (pp. 437–470). Oxford, UK: Oxford University Press:.

MacKenzie, I. S. (2002). Introduction to this special issue on text entry for mobile computing. *Human-Computer Interaction, 17*, 141–145.

MacKenzie, I. S., & Guiard, Y. (2001). The two-handed desktop interface: Are we there yet? *Proceedings from the ACM CHI 2001 Conference on Human Factors in Computing Systems, Extended Abstracts*, 351–352.

MacKenzie, I. S., Kober, H., Smith, D., Jones, T., & Skepner, E. (2001). LetterWise: Prefix-based disambiguation for mobile text input. *Proceedings from the ACM UIST 2001 Symposium on User Interface Software & Technology*, 111–120.

MacKenzie, I. S., & Oniszczak, A. (1998). A comparison of three selection techniques for touchpads. *Proceedings from the ACM CHI '98 Conference on Human Factors in Computing Systems*, 336–343.

MacKenzie, I. S., Sellen, A., & Buxton, W. (1991). A Comparison of input devices in elemental pointing and dragging tasks. *Proceedings from ACM CHI '91 Conference on Human Factors in Computing Systems*, 161–166.

MacKenzie, I. S., & Soukoreff, R. W. (2002). A model of two-thumb text entry. *Proceedings of the Graphics Interface*, 117–124.

MacKenzie, I. S., & Zhang, S. (1997). The immediate usability of graffiti. *Proceedings of the Graphics Interface '97*, 129–137.

Maes, P., Darrell, T., Blumberg, B., & Pentland, A. (1997). The ALIVE system: Wireless, full-body interaction with autonomous agents. *Multimedia Syst. 5*, 2 (Mar. 1997), 105–112.

Marklin, R., Simoneau, G., & Monroe, J. (1997). The effect of split and vertically-inclined computer keyboards on wrist and forearm posture. *Proceedings of the HFES Human Factors and Ergonomics Society 41st Annual Meeting*, 642–646.

Matsushita, N., & Rekimoto, J. (1997). Holo Wall: Designing a finger, hand, body, and object sensitive wall. *Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology*, 209–210.

McLoone, H., Hinckley, K., & Cutrell, E. (2003). Bimanual interaction on the Microsoft Office keyboard. *Proceedings from the INTERACT 2003*.

Microsoft Corp. (2002). Windows XP Pointer Ballistics (January 17, 2006). http://www.microsoft.com/whdc/device/input/pointerbal.mspx

Mine, M., Brooks, F., & Sequin, C. (1997). Moving objects in space: Expoiting proprioception in virtual-environment interaction. *Computer Graphics, 31* (Proc. SIGGRAPH'97): 19–26.

Moran, T., Chiu, P., & van Melle, W. (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. *Proceedings from the UIST '97*, 45–54.

Myers, B., Lie, K., & Yang, B. (2000). Two-handed input using a PDA and a mouse. *Proceedings from the CHI 2000*, 41–48.

Myers, B., Stiel, H., & Gargiulo, R. (1998). Collaboration using multiple PDAs connected to a PC. *Proceedings from the ACM CSCW '98 Conference on Computer Supported Cooperative Work*, 285–294.

Norman, D. (1990). *The design of everyday things*. New York: Doubleday.

Oakley, I., Brewster, S., & Gray, P. (2001). Solving multi-target haptic problems in menu interaction. *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems, Extended Abstracts*, 357–358.

Olsen, D. R., & Nielsen, T. (2001). Laser pointer interaction. *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems*, 17–22.

Olson, J. R., & Olson, G. M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction, 5*(2–3), 221–266.

Oviatt, S., DeAngeli, A., & Kuhn, K. (1997). Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of the CHI '97*, 415–422.

Poupyrev, I., & Maruyama, S. (2003). Tactile interfaces for small touch screens. *Proceedings of the UIST 2003 Symposium on User Interface Software & Technology*, 217–220.

Putz-Anderson, V. (1988). *Cumulative trauma disorders: A manual for musculoskeletal diseases of the upper limbs*. Bristol, PA: Taylor & Francis.

Raskar, R., Beardsley, P., van Baar, J., Wang, Y., Dietz, P. H., Lee, J., Leigh, D., & Willwacher, T. (2004). "RFIG Lamps: Interacting with a Self-Describing World via Photosensing Wireless Tags and Projectors", *ACM Transactions on Graphics (TOG) SIGGRAPH, 23*(3), 406–415.

Raskin, J. (2000). *The humane interface: New directions for designing interactive systems*, ACM Press, New York.

Rekimoto, J. (1996). Tilting operations for small screen interfaces. *Proceedings of the ACM UIST '96 Symposium on User Interface Software & Technology*, 167–168.

Rekimoto, J. (1998). A multiple device approach for supporting whiteboard-based interactions. *Proceedings of the CHI '98*, 344–351.

Rekimoto, J., Ayatsuka, Y., & Kohno, M. (2003). SyncTap: An interaction technique for mobile networking. *Proceedings of the Mobile HCI 2003*, 104–115.

Rempel, D., Bach, J., Gordon, L., & Tal, R. (1998). Effects of forearm pronation/supination on carpal tunnel pressure. *J Hand Surgery, 23*(1), 38–42.

Rime, B., & Schiaratura, L. (1991). Gesture and speech. In *Fundamentals of Nonverbal Behaviour* (pp. 239–281). New York: Press Syndicate of the University of Cambridge.

Rutledge, J., & Selker, T. (1990). Force-to-motion functions for pointing. *Proceedings of the Interact '90: The IFIP Conference on Human-Computer Interaction*, 701–706.

Saund, E., & Lank, E. (2003). Stylus input and editing without prior selection of mode. *Proceedings of the UIST '03*, 213–216.

Sawhney, N., & Schmandt, C. M. (2000). Nomadic radio: Speech and audio interaction for contextual messaging in nomadic environments. *ACM Transactions on Computer-Human Interaction, 7*(3), 353–383.

Schilit, B. N., Adams, N. I., & Want, R. (1994). Context-aware computing applications. *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 85–90.

Schilit, B. N., Golovchinsky, G., & Price, M. N. (1998). Beyond paper: Supporting active reading with free form digital ink annotations. *Proceedings of the CHI'98*, 249–256.

Schmandt, C. M., Marmasse, N., Marti, S., Sawhney, N., & Wheeler, S. (2000). Everywhere messaging. *IBM Systems Journal, 39*(3–4), 660–686.

Scott, S. D., Grant, K. D., & Mandryk, R. L. (2003). System guidelines for co-located, collaborative work on a tabletop display. *Proceedings of the ECSCW '03, European Conference Computer-Supported Cooperative Work*, 159–178.

Sears, A., Plaisant, C., & Shneiderman, B. (1992). A new era for high precision touchscreens. In H. Hartson, & D. Hix (Eds.), *Advances in Human-Computer Interaction* (pp. 1–33). Ablex Publishers, Norwood, NJ.

Sears, A., & Shneiderman, B. (1991). High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies, 34*(4), 593–613.

Sellen, A., Kurtenbach, G., & Buxton, W. (1992). The prevention of mode errors through sensory feedback. *J. Human Computer Interaction, 7*(2), 141–164.

Sellen, A. J., & Harper, H. R. (2002). *The myth of the paperless office*. Cambridge, MA: MIT Press.

Shen, C., Everitt, K., & Ryall, K. (2003). UbiTable: Impromptu face-to-face collaboration on horizontal interactive surfaces. *Proceedings of the UbiComp 2003.*

Smith, M. A., Davenport, D., & Hwa, H. (2003). AURA: A mobile platform for object and location annotation. *Proceedings of the UbiComp 2003.*.

Smutz, W., Serina, E., Bloom, T., & Rempel, D. (1994). A system for evaluating the effect of keyboard design on force, posture, comfort, and productivity. *Ergonomics, 37*(10), 1649–1660.

Snibbe, S. S., MacLean, K. E., Shaw, R., Roderick, J., Verplank, W. L., & Scheeff, M. (2001). Haptic techniques for media control. In *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology* (Orlando, Florida, November 11–14, 2001). *UIST '01* (199–200) ACM Press, New York, NY.

Sommer, R. (1965). Further studies of small group ecology. *Sociometry, 28*, 337–348.

Stifelman, L. (1996). Augmenting real-world objects: A paper-based audio notebook. *CHI'96 Conference Companion*.

Swaminathan, K., & Sato, S. (1997). Interaction design for large displays. *Interactions 4*, 1 (Jan. 1997), 15–24.

Trevor, J., Hilbert, D. M., Schilit, B. N., & Koh, T. K. (2001). From desktop to phonetop: A UI for web interaction on very small devices. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, 121–130.

Trimble, J., Wales, R., & Gossweiler, R. (2003). NASA's MERBoard. In K. O'Hara, M. Perry, E. Churchill, & D. Russell (Eds.), *Public and Situated Displays: Social and interactional aspects of shared display technologies*. Kluwer.

Viviani, P., & Flash, T. (1995). Minimum-jerk, two-thirds power law and isochrony: Converging approaches to the study of movement planning. *Journal of Experimental Psychology: Perception and Performance, 21*, 32–53.

Vogel, D., & Balakrishnan, R. (2004). Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. *Proceedings from the UIST 2004*.

Vogel, D., & Balakrishnan, R. (2005). Distant freehand pointing and clicking on very large, high resolution displays. *UIST 2005*, 33–42.

Want, R., & Borriello, G. (2000, May–June). Survey on information appliances. *IEEE Personal Communications*, 24–31.

Want, R., Fishkin, K. P., Gujar, A., & Harrison, B. L. (1999). Bridging physical and virtual worlds with electronic tags. *Proceedings of the ACM CHI '99 Conference on Human Factors in Computing Systems*, 370–377.

Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active bage location system. *ACM Transactions on Information Systems, 10*(1), 91–102.

Weiser, M. (1991, September). The computer for the 21st century. *Scientific American*, 94–104.

Wherry, E. (2003). Scroll ring performance evaluation. *CHI '03 Extended Abstracts,* ACM, New York, 758–759.

Wickens, C. (1992). *Engineering psychology and human performance*. New York: HarperCollins.

Wigdor, D., & Balakrishnan, R. (2003). TiltText: Using tilt for text input to mobile phones. *ACM UIST '03 Symposium on User Interface Software & Technology*, 81–90.

Wilson, A. (2004). TouchLight: An imaging touch screen and display for gesture-based interaction. *International Conference on Multimodal Interfaces (ICMI)*.

Wilson, A. D. (2005). PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA, October 23–26, 2005). UIST '05. ACM Press, New York, NY, 83–92.

Wilson, A., & Oliver, N. (2003). GWindows: Towards robust perception-based UI. *Proceedings of CVPR 2003 (Workshop on Computer Vision for HCI)*.

Wilson, F. R. (1998). *The hand: How its use shapes the brain, language, and human culture.* New York: Pantheon Books.

Wolf, C., Rhyne, J., & Ellozy, H. (1989). The paper-like interface. In G. Salvendy, & M. Smith (Eds.), *Designing and using human computer interface and knowledge-based systems* (pp. 494–501). Amsterdam, Elsevier.

Wu, M., & Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of the UIST 2003*, 193–202.

Yamada, H. (1980). A historical study of typewriters and typing methods: From the position of planning Japanese parallels. *J. Information Processing, 24*(4), 175–202.

Yee, K.-P. (2003). Peephole displays: Pen interaction on spatially aware handheld computers. *Proceedings of the CHI 2003*.

Yee, K.-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. *Proceedings of the CHI 2003*, 401–408.

Zhai, S. (1993). Human performance evaluation of manipulation schemes in virtual environments. *Proceedings of the IEEE Virtual Reality International Symposium (VRAIS '93)*, 155–161.

Zhai, S. (1998). User performance in relation to 3D input device design. *Computer Graphics, 32*(4), 50–54.

Zhai, S., Hunter, M., & Smith, B. A. (2000). The metropolis keyboard—An exploration of quantitative techniques for virtual keyboard design. *CHI Letters, 2*(2), 119–128.

Zhai, S., & Milgram, P. (1993). Human performance evaluation of isometric and elastic rate controllers in a 6DoF tracking task. *Proceedings of the SPIE Telemanipulator Technology.*

Zhai, S., Milgram, P., & Buxton, W. (1996). The influence of muscle groups on performance of multiple degree-of-freedom input. *Proceedings of CHI '96: ACM Conference on Human Factors in Computing Systems*, 308–315.

Zhai, S., Smith, B. A., & Selker, T. (1997). Improving browsing performance: A study of four input devices for scrolling and pointing tasks. *Proceedings of the INTERACT97: The Sixth IFIP Conference on Human-Computer Interaction*, 286–292.

Zhao, S., & Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. *Proceedings from the UIST 2004*, 33–42.

◆10◆

SENSOR- AND RECOGNITION-BASED INPUT FOR INTERACTION

Andrew D. *Wilson*
Microsoft Research

Introduction **178**
Sensors and Sensing Modes **178**
Occupancy and Motion 179
Range Sensing 179
Position 179
Movement and Orientation 180
Touch 180
Gaze and Eyetracking 181
Speech 181
Gesture 181
Identity 182
Context 182
Affect 182
Brain Interfaces 183
Signal Processing **183**
Preprocessing 183
Feature Selection 184
Classification and Modeling 185
An Example System **187**
Considerations in Designing Recognition-Based Systems **190**
Computational Cost, Utility, and the Cost of Failure . . 190
The Role of Feedback 191
Implicit and Explicit Interaction 191
The Importance of Context 191
The Importance of *A Priori* Knowledge 192
Generalize or Specialize? 193
Traditional vs. Nontraditional Interfaces 193
Evaluating Novel Interfaces 194
Conclusion **195**
Acknowledgments **195**
References **196**

INTRODUCTION

Sensors convert a physical signal into an electrical signal that may be manipulated symbolically on a computer. A wide variety of sensors have been developed for aerospace, automotive and robotics applications (Fraden, 2003). Continual innovations in manufacturing and reductions in cost have allowed many sensing technologies to find application in consumer products. An interesting example is the development of the ubiquitous computer mouse. Douglas Engelbart's original mouse, so named because its wire "tail" came out of its end, used two metal wheels and a pair of potentiometers to sense the wheels rolling over a desk surface. Soon, mice used a ball and a pair of optical encoders to convert the movement of the hand into digital signals indicating precise relative motion. Now, even the most inexpensive mice employ a specialized camera and image processing algorithms to sense motions at the scale of one one-thousandth of an inch several thousand times per second. *Accelerometers*, devices that sense acceleration due to motion and the constant acceleration due to gravity, are another interesting example. Today's tiny accelerometers were originally developed for application in automotive air-bag systems. Digital cameras now incorporate accelerometers to sense whether a picture is taken in landscape or portrait mode, and save the digital photo appropriately. Many laptops with built-in hard disks also include accelerometers to detect when the laptop has been dropped, and park the hard drive before impact. Meanwhile, mobile phone manufacturers are experimenting with phones that use accelerometers to sense motion for use in interaction, such as in-the-air dialing, scrolling, and detecting the user's walking pattern.

Research in human-computer interaction (HCI) explores the application of sensing to enhance interaction. The motivation of this work is varied. Some researches seek to expand the array of desktop input options, or to build completely new computing form factors such as mobile devices that know where they are pointed and intelligent environments that are aware of their inhabitants. Other researchers are interested in using sensors to make our machines behave more like we do, or alternatively to make them complement human abilities. Entertainment, surveillance, safety, productivity, mobile computing, and affective computing are all active areas in which researchers are applying sensors in interesting ways.

While a wide array of sensors is available to researchers, rarely does a sensor address exactly the needs of a given application. Consider building into a computer the capability to sense when its user is frustrated. Detection of user frustration would allow a computer to respond by adopting a new strategy of interaction, playing soothing music, or even calling technical support; however, today, no "frustration meter" may be purchased at the local electronics store. What are the alternatives? A microphone could be used to sense when the user mutters or yells at the machine. A pressure sensor in the mouse and keyboard could detect whether the user is typing harder or squeezing the mouse in frustration (Klein, Moon, & Picard, 2002; Reynolds, 2001). A webcam might detect scowling or furrowing of the eyebrows. Sensors in the chair could detect user agitation (Tan, Slivovsky, & Pentland, 2001). Ultimately, the system chosen should probably exploit a consistent, predictable relationship between the output of one of these sensors and the user's frustration level; for example, if the mouse is squeezed at a level exceeding some set threshold, the computer may conclude that the user is frustrated.

In our effort to build a frustration detector, we may find a number of issues confounding the relationship between the sensors and the state to be detected:

- There is no easy a priori mapping between the output of the sensors and the presumed state of frustration in the user. Implementation of a pressure sensor on the mouse requires observation of the user over time to determine how much pressure reliably indicates frustration. Implementation of the more complex approach of detecting furrowed brows by computer vision requires an elaborate image processing algorithm.
- The output of the sensors is noisy and often accompanied by a degree of uncertainty.
- Initial experimentation reveals that while no single sensor seems satisfactory, it sometimes may suffice to combine the output of multiple sensors.
- Our preconceived notions of frustration may not correspond to what the sensors observe. This may cause us to revisit our understanding of how people express frustration, which, in turn, may lead us to a different choice of sensors.
- The manner in which the user expresses frustration depends greatly on the user's current task and other contextual factors, such as the time of day and level of arousal. Exploiting knowledge of the user's current application may address many cases where our algorithm for detecting frustration fails.
- After realizing that our frustration detector does not perform flawlessly, we struggle to balance the cost of our system making mistakes with the benefit the system provides.

These are just some of the considerations that are typical in a nontrivial application of sensors to recognition in HCI. While this article does not propose to solve the problem of detecting and responding to user frustration, it will survey aspects of sensor-based recognition highlighted by this example. In particular, this article presents the variety of available sensors and how they are often used in interactive systems. Signal processing, recognition techniques, and further considerations in designing sensor and recognition-based interactive systems are briefly addressed.

SENSORS AND SENSING MODES

This article focuses only on those sensors relevant to interactive applications and their typical modes of use. Experimenting with such sensors has never been easier. Microcontrollers such as the Microchip PIC and BasicStamp can interface sensors to PCs and other devices, and can be programmed using high-level languages such as C and BASIC. The Phidgets hardware toolkit (Greenberg & Fitchett, 2001) enables effortless "plug and play"

prototyping with modular sensors and actuators that are plugged into a base interface board, and provides software APIs. The Berkeley and Intel Mote projects also offer wireless sensor packages useful in data collection and sensor networks (Kahn, Katz, & Pister, 2000; Nachman, Kling, Huang, & Hummel, 2005). Another source for inexpensive sensors and sensor interface kits is the hobbyist robotics community.

Occupancy and Motion

Probably owing to the importance of sensing technology in security applications, many devices and techniques exist to sense either motion or a person's presence (occupancy). Among these are

- Air pressure sensors that detect changes in air pressure resulting from the opening of doors and windows.
- Capacitive sensors that detect capacitance changes induced by the body.
- Acoustic sensors.
- Photoelectric and laser-based sensors that detect disruption of light.
- Optoelectric sensors that detect variations in illumination.
- Pressure mat switches and strain gauges.
- Contact and noncontact (magnetic) switches.
- Vibration detectors.
- Infrared motion detectors.
- Active microwave and ultrasonic detectors.
- Triboelectric detectors that detect the static electric charge of a moving object (Fraden, 2003).

Perhaps one of the most familiar motion detectors is the *passive infrared* (PIR) detector, which is sensitive to small changes in the pattern of infrared radiation within the spectral range of 4 to 20 micrometers ("far infrared"). PIR detectors sense heat changes over a small duration of time, indicating the presence of a person moving through the room. These devices often control lights in office buildings, and can be useful in office-awareness applications when combined with other sensors.

More selective motion and occupancy detection can be obtained with video cameras and simple computer vision techniques. For example, a computer vision system allows for the definition of multiple *regions of interest* (ROI) that allow fine distinctions regarding the location of motion. Such a system may thus be able to ignore distracting motion.

Range Sensing

Range sensors calculate the distance to a given object. Such detectors can be used as occupancy detectors, and are also useful in motion- and gesture-driven interfaces.

Many range and proximity sensors triangulate the position of the nearest object. For example, the Sharp IR Ranger emits a controlled burst of near infrared light from a light emitting diode (LED). This light is reflected by any object within a few feet and is focused onto a small linear *charge-coupled devices* (CCD) array that is displaced slightly from the emitter. The position of the reflection on the sensor can be related the distance to the object by trigonometry. Similar approaches can be used over a longer effective distance with the use of lasers rather than LEDs.

Stereo computer vision systems similarly use triangulation to calculate depth. If the same object is detected in two displaced views, the difference in their sensed 2D positions, called "disparity," can be related to the depth of the object (Forsyth & Ponce, 2002; Horn, 1986). Stereo vision techniques may be used to determine the depth of a discrete object in the scene, or to compute depth at each point in the image to arrive at a full range image.

A second approach to calculating range is based on measuring the time-of-flight of an emitted signal. The Polaroid ultrasonic ranging device, for example, was originally developed for auto-focus cameras, and subsequently became popular in robotics. Such sensors emit a narrow ultrasonic "chirp" and later detect the chirp's reflection. The duration in time between the chirp and the detection of the reflection is used to calculate the distance to the object. Ultrasonic range finders can sometimes be confused by multiple reflections of the same chirp; such difficulties are eliminated by measuring the time of flight of emitted light rather than sound, but such sensors are still comparatively exotic.

Position

Designers of sensing-based interactive systems would probably most like a low-power, wireless, inexpensive 3D position sensor that does not rely on the installation of complicated infrastructure. Originally designed for military application, *Global Positioning Satellite* (GPS) devices are useful for sensing street-level movement but are limited to outdoor application. Unfortunately, no indoor tracking standard has gained the popularity of GPS.

The position of a wireless RF receiver can be determined by measuring signal strengths to RF transmitters of known position using Wi-Fi, Bluetooth, and GSM standards (LaMarca et al., 2005). Under the assumption that signal strength approximates distance, position may determined by triangulation, but often interference from buildings, walls, furniture, and even people can be troublesome. Another approach is to treat the pattern of signal strengths as a "signature" which, when recognized later, indicates the position associated with the signature (Krumm & Horvitz, 2004). Using Wi-Fi transceivers and a number of Wi-Fi access points, position can be calculated within several feet of accuracy under ideal laboratory conditions using a combination of approaches (Letchner, Fox, & LaMarca, 2005). Finally, the Ubisense location system achieves accuracy on the order of 15 cm indoors by using arrival time and angle of RF signals in ultrawideband (UWB) frequencies.

Commercially available motion capture systems employ a variety of strategies to track the position and orientation of multiple points. These systems are generally used to record human motions for applications such as video game character animation. Most require the performer to wear several small tracking

devices. For example, when precisely calibrated, the Polhemus and Ascension magnetic tracking devices achieve millimeter accurate, six degree of freedom (position and orientation) tracking of multiple points, but rely on technology that connects each tracking device to a base station with wires. Such products have been very useful in prototyping gesture-based interactions that require accurate 3D position and orientation (Fitzmaurice, Ishii, & Buxton, 1995; Hinckley, Pausch, Proffitt, & Kassel, 1998; Ware, 1990; Ware & Jessome, 1988), but often are too expensive for widespread use.

Much research in computer vision focuses on accurate and reliable object tracking. Where multiple cameras are available, it is possible to compute the 3D position of a tracked object using triangulation. To recover 3D position useful for interactive applications in a typical room setting, such cameras require careful calibration. Several prototype interactive systems use vision techniques to track the hands, head, and body of a user. Often, a model of shape and appearance that includes typical skin color is used to track the head and hands. The ALIVE system, for example, determines the 2D position of the head and hands of a user by first extracting the user's silhouette against a controlled (static) background. Point of high curvature along this contour are then extracted and tracked as hands or head. Later variants exploit color information as well. Depth is computed by assuming the user is standing on a flat floor (Maes, Darrell, Blumberg, & Pentland, 1995; Wren, Azarbayejani, Darrell, & Pentland, 1995).

Computer vision-based tracking systems often suffer from poor tracking reliability and sensitivity to variations in background illumination. Tracking reliability can be enhanced by controlling the appearance of the object so that it can be tracked unambiguously. A number of professional motion capture systems, such as the Vicon Peak system, rely on passive, wireless, infrared-reflective pieces, but also require a powerful infrared light source and multiple, redundant sensors (cameras) to minimize missing data resulting from occlusion. Alternatively, cameras sensitive in the infrared domain can be used to track an infrared LED (IR-LED). The *position sensitive device* (PSD), for example, is an inexpensive, camera-like device that reports the brightest spot on its imaging array and is thus suitable for inexpensive IR-LED-based tracking systems. Multiple IR-LEDs can be tracked using a PSD by carefully controlling when each is illuminated. Gross room-level location can be determined using the simplest infrared detectors and IR-LEDs that transmit the identify of the user by blinking specific patterns over time, much like a television remote control (Want, Hopper, Falcao, & Gibbons, 1992).

Acoustic tracking systems are able to triangulate position using time-of-flight measurements. One approach is to equip the room with multiple detectors that are able to hear a mobile tracking device equipped to emit a sound at a known frequency. This configuration can also be inverted, with the detector on the tracked device and the emitters in the environment (Smith, Balakrishnan, Goraczko, & Priyantha, 2004; Ward, Jones, & Hopper, 1997). Related signal processing algorithms can combine the output of two or more microphones to triangulate the position of an arbitrary sound source (Rui & Florencio, 2003). This approach can be particularly effective when combined with other techniques such as computer vision-based face tracking (Zhang & Hershey, 2005).

Movement and Orientation

Unlike most tracking technologies, a number of movement and orientation sensors do not rely on external infrastructure. *Inertial* sensors, for example, sense spatial and angular motion (translation and rotation). They can be used for activity recognition as well as gesture and body motion-based interactive applications where it is acceptable to wear or hold a small wireless sensor package (Bao & Intille, 2004; Hinckley, Pierce, Sinclair, & Horvitz, 2000; Lee & Mase, 2002).

Very simple tilt sensors such as mercury switches have been used for years to sense gross orientation. More recently, inexpensive accelerometer devices using *micro-electro-mechanical systems* (MEMS) technology were developed for application in automotive airbag systems (Kovacs, 1998). MEMS accelerometers feature a tiny proof mass or cantilever beam and deflection sensing circuitry to sense both varying accelerations due to movement as well as the constant acceleration due to gravity. Two-axis MEMS accelerometers can be applied to sensing tilt (pitch and roll) and have been used in gaming controllers. But nonmilitary accelerometers are not sufficiently precise to support the double integration of acceleration necessary to calculate position information for more than a few seconds. In such applications, it may suffice to add a coarse position sensor to combat the effects of drift.

MEMS technology has also allowed the development of *gyroscope* devices that sense angular acceleration rather than absolute orientation. These devices have been used in stabilizing handheld cameras and in the Gyromouse product, which maps relative change in gyroscope orientation to the relative motion of the mouse cursor.

Magnetometers are compact, solid-state devices able to detect the strength of the earth's magnetic field along its principle axis, and so are useful in determining absolute orientation information. The output of a pair of orthogonally mounted magnetometers held level may be combined to find magnetic north. It is common to combine a two-axis accelerator with a two-axis magnetometer to "correct" the output of the magnetometer when it is not held level (Caruso, 1997). Three axis magnetometers are available, but alone do not give a true 3D orientation (e.g., a magnetometer's reading does not change when it is rotated about magnetic north).

Touch

The microswitch typical of today's mouse requires a certain amount of force to activate, thus allowing a user to comfortably rest their forefinger on the button without accidentally clicking. Pressure sensors, on the other hand, sense a continuous range of pressure states. Historically, these have been useful in robotics, where they play an important role in designing control systems for manipulators. *Polyvinylidene fluoride* (PVDF) films and *force sensitive resistors* (FSRs) are two inexpensive types of pressure sensors with good dynamic range and form factors useful for small devices and interactive systems. Flexible *strain gauges* utilizing the piezoresistive effect have a resistance related to the amount of deformation ("bend") applied to the sensor. Such gauges have been used as the basis for inexpensive

glove devices that sense the deflection of each of the fingers of the hand.

Capacitive sensing is based on the property that nearly any object is capable of storing electric charge, and that charge will flow between two objects when touching or in close proximity. A zero-force touch sensor can be implemented with a *charge transfer* technique, whereby the capacitance of an electrode is estimated by measuring the time taken for an electrode to discharge a small applied charge (Hinckley & Sinclair, 1999). This time drops dramatically when the user places a finger on the electrode, since the user's body takes on much of the charge. Other capacitive sensing techniques can sense an object before it touches the electrode, making them suitable as proximity sensors for a wide variety of interactive applications (Baxter, 1996; Smith, 1999; Vranish, McConnell, & Mahalingam, 1991). Multiple electrodes can be used to implement position sensitive sliders, such as the wheel on the Apple iPod.

Most common touch screens report the single 2D position of the user's finger touching or pressing the screen (Sears, Plaisant, & Shneiderman, 1992). *Resistive* touch screens use two large transparent conductive overlays that vary in resistance over their length. When the user presses on the screen, the overlays are brought into contact, and a voltage applied to one or the other overlay is used to detect the horizontal or vertical position of the finger. *Capacitive* touch screens use capacitive sensing to sense touch, and the relative difference in the charge sensed at each corner of the screen to determine position. Recently, the ability to more precisely sense the touch location and also sense the area and shape of the touching object has been enabled by embedding multiple capacitive sensors in the display surface (Dietz & Leigh, 2001; Rekimoto, 2002). Finally, *surface acoustic wave* systems rely on sensing the finger's disruption of surface acoustic waves applied to a screen surface (Pickering, 1986).

Finally, computer vision techniques have been applied to sense touch (Fails & Olsen, 2002; Han, 2005; Matsushita & Rekimoto, 1997; Smart Technologies, Inc., 2007; Tomasi, Rafii, & Torunoglu, 2003; Wellner, 1993; Wilson, 2005). Using computer vision to sense touch over an area has a number of advantages: first, these techniques usually do not require a special instrumented surface as do most touch screens. Secondly, vision techniques naturally support multiple touch points. Finally, vision techniques enable the ability to detect and recognize a variety of objects besides fingers. For example, barcode-like visual codes may be applied to uniquely identify objects such as game pieces placed on a surface.

Gaze and Eyetracking

Gaze detection refers to determining where a person is looking and is principally the domain of computer vision. It is possible to very coarsely determine head orientation using techniques related to face detection (Wu, Toyama, & Huang, 2000), but head orientation is often a poor indicator of where someone is looking. The goal of *eyetracking* systems is precisely determine where the user is looking, or *foveating*. Usually, these techniques are based on precise tracking of multiple reflections of an infrared illuminant off the eye's cornea. For good performance, however, eyetracking systems require careful per-user calibration, and so have seen limited general application in interactive systems (Beymer & Flickner, 2003; Jacob, 1993; Tobii, 2005; Zhai, Morimoto, & Ihde, 1999).

Rather than determining gaze in a general fashion only to later match the gaze direction to one of several known objects, an alternative is to determine only whether user is looking at the object.. The detector can then be embedded in the object itself. Furthermore, the reflection of an infrared illuminant by the cornea and retina can be detected by simple image processing techniques when the camera and infrared illuminant are colocated (Haro, Flicker, & Essa, 2000; Shell et al., 2004).

Speech

The long history of research on speech recognition techniques has resulted in commodity systems that bring modern speech recognition to anyone with a PC and an inexpensive microphone (Rabiner & Juang, 1993). New interactive systems, however, highlight the need for further work. Current systems, for example, function poorly without a "close-talk" microphone and in noisy environments, and so are unsuited for use in such contexts as intelligent rooms and mobile scenarios.

The *array microphone* combines audio from multiple microphones to address the problems of multiple sound sources and noisy environments. Through the process of *beamforming*, the outputs of the multiple microphones of an array is combined to form a single audio signal in which all but the dominant speaker's signal has been removed. Beamforming can also reveal information about the position of the speaker (Tashev & Malvar, 2005).

To achieve robustness, speech may also be combined with other input modalities such as pen gestures (Oviatt, 2002). Such approaches usually require a sophisticated model of the user's interaction. Perhaps inspired by HAL in *2001: A Space Odyssey*, some researchers have proposed incorporating computer vision-based lip-reading techniques into the speech interpretation process (Stork, 1998). Finally, information such as intonation, prosody and conversational turn taking can be valuable in interactive systems (Bilmes et al., 2005; Choudhury & Pentland, 2003; Pentland, 2004).

Gesture

Many notions of *gesture* exist in interactive systems, and thus many sensor systems are applicable. A gesture can be thought of as a specific hand pose, a spatial trajectory of the hands or stylus, pointing or other motion to indicate an object, or the quality of a motion of almost any body part as it relates to a given application context (McNeill, 1992).

Many of the previously mentioned tracking and movement sensing technologies have been applied to sense and recognize gestures. For example, a wireless sensor package with multiple accelerometers or gyros can capture motion information useful in recognizing many gestures. Computer vision techniques also can be used to track body parts such as the hands

and head, as well as overall motion qualities that can be interpreted as gesture. Often such systems ease the sensing task by requiring the user to wear brightly colored gloves, or by training precise models of skin color (Brashear, Starner, Lukowicz, & Junker, 2003).

Pen gestures are often studied on the tablet computer, a form factor that often uses the Wacom electromagnetic positioning technology. This system uses coils embedded in the pen and under the display to find pen position, limited height (hover) above the surface, tilt, pressure, and button state. It can also support multiple simultaneous inputs. The gestures themselves are usually modeled as simple "flick" gestures or spatial trajectories.

While computer vision techniques have been explored to recover detailed hand pose information, gloves with built-in sensors are more commonly worn for this purpose (Baudel & Beaudouin-Lafon, 1993). Early *virtual reality* (VR) systems, for example, used magnetic trackers attached to gloves equipped with bend sensors to recover the position, orientation, and pose of the hands. More recently, vision-based professional motion capture systems that track infrared retro-reflective balls have been used in a similar fashion. With such precise hand shape information, it is possible to point at an object with the index finger, and then make a motion similar to pulling the trigger of a gun to effect an action (Vogel & Balakrishnan, 2005).

Identity

In interactive systems, it is often useful to know the identity of an object or user, and a variety of sensing systems are designed to recognize known objects. Object recognition is an active research area in computer vision. There are practical techniques for quickly recognizing one of many known flat objects such as photos or book covers, for example (Lowe, 2004). Computer vision-based face recognition techniques have also been shown to work in fairly controlled settings (Li & Jain, 2005). General object recognition and face recognition in uncontrolled settings is still difficult, however.

Beyond face recognition, biometrics uses a variety of sensing technologies. Fingerprint recognition hardware, for example, uses optical scanning technology, or an array of tiny capacitive sensors, to construct an overall picture of the fingerprint. Since Johansen's early experiments demonstrating an ability to recognize human motion from point-light displays (Johansson, 1973), researchers have worked on gait recognition techniques from video (Boyd & Little, 2005). Iris, retina, hand geometry, vascular pattern, handwritten signature, and voice dynamics are other biometric techniques employing sensing technology (Sugiura & Koseki, 1997; Wayman, Jain, Maltoni, & Maio, 2004).

In the absence of reliable techniques to recognize an object by its natural properties, it is often useful to "tag" an object with a standard, easily recognizable marker that reveals the object's identity. Visual codes such as the ubiquitous UPC bar code symbols, for example, are read by laser scanning systems, which are now small enough to be incorporated into mobile devices. Two-dimensional "matrix codes" such as the QR code pack more bits into the same space, and have been used in a variety of interactive systems that recognize them by image analysis (Kato, Billinghurst, Poupyrev, Imamoto, & Tachibana, 2000; Rekimoto & Ayatsuka, 2000).

Recently, *radio frequency identification* (RFID) tags have gained in popularity. RFID tags themselves are usually passive, can be made small and unobtrusive, are cheap to manufacture, and can be read at a distance. A scanning antenna that emits an RF signal reads the tags; this signal, in turn, powers the tags with enough energy to respond with an identification code. RFID systems vary in terms of range, power requirements, antenna and tag form factors, bit depth, and so on (Garfinkel & Rosenberg, 2005).They are thus particularly attractive for commercial inventory management applications and have been applied to interactive systems (Want, Fishkin, Gujar, & Harrison, 1999).

Context

Sensors can provide important information about the *context* of the user or device. For example, a computer may listen in on an office to determine whether a meeting is in progress, and if so withhold noncritical notifications (Oliver & Horvitz, 2005). A later section explores the role of context in interpreting the output of sensors.

Simple context sensors are especially useful in mobile applications. Environmental sensors that detect such information as air temperature, lighting quality, and air pressure may be more directly relevant to the application than the absolute location given by a GPS sensor (Lester et al., 2006; Schmidt, Beigl, & Gellersen, 1999). Context sensors may be used to determine the user's *activity*, or what the user is currently doing. In mobile applications, an inertial sensor may be used to determine the current transportation mode of the user, while a microphone may be used to conclude that the user is engaged in a conversation. An array of simple switches placed throughout a household environment, such as on kitchen cabinets and drawers, may be all that is needed to reliably determine the activities of its inhabitants (Tapia, Intille, & Larson, 2004; Wilson & Atkeson, 2005).

Affect

In psychology, *affect* refers to an emotion or subjective feeling. Recently there has been interest in applying sensing technology to allow interactive systems to respond appropriately to (and perhaps influence) the user's affect (Picard, 2000). A system might respond to the user's boredom, interest, pleasure, stress, or frustration (as in the example in the introduction) by changing aspects of the interaction.

Like other multimodal systems, an *affective computing* system is likely to integrate a variety of conventional sensors. There is an emphasis, however, on the use of *physiological sensors* to recover physical data that may be related to the user's affective state. For example, the *galvanic skin response* (GSR) sensor measures the skin's conductivity, which increases quickly when the user is startled or experiences anxiety. The *blood volume pulse* (BVP) sensor measures blood pressure over a local region by measuring the reflectance of a bright infrared light, and can detect certain states of arousal when applied to the fingertips. Respiration rate can be sensed by measuring the amount of

stretch in an elastic band worn around the chest. The *electromyogram* (EMG) sensor measures the amount of electrical activity produced when the muscle it is placed over contracts, and is useful in detecting jaw clenching, and contraction of various muscles related to facial expressions. Finally, the *electrocardiogram* (ECG or EKG) measures heart rate.

In designing affective computing systems, it is often difficult to determining the mapping of sensor outputs to application specific quantities, such as emotional state. Particularly challenging is the task of identifying specific physical correlates for broadly defined emotional states such as "frustration." Finally, physiological sensors are unsuitable for many applications because of the difficulty in deploying them: many must be placed on particular locations on the body, may require good contact with the skin, and are susceptible to differences among individual users or even the same user from day to day.

Brain Interfaces

Advances in cognitive neuroscience and brain imaging technology have spurred initial explorations into interfacing computers directly with a user's brain activity. Much of the work is motivated by a desire to help individuals who have lost the motor skills necessary to use traditional interfaces. Thus, the goal of *brain-computer interfaces* (BCI) is often to enable users to explicitly manipulate brain activity in order to provide input to a system. Such interfaces typically emulate traditional interfaces by triggering keystrokes and cursor control. However, future applications will likely take advantage of the unique abilities of BCI systems to enable completely new styles of interaction (Hjelm & Browall, 2000).

BCI is generally limited to brain imaging techniques that are noninvasive and do not require bulky, expensive equipment. The *electroencephalograph* (EEG) measures electrical activity at local parts of the brain using electrodes placed carefully on the scalp. EEG has low spatial resolution compared to other brain imaging techniques, but has relatively good temporal resolution. *Functional near infrared* (fNIR) imaging measures blood flow in local regions of the brain by calculating the absorption of infrared light directed into the scalp. The technology suffers, however, from low temporal resolution, but obtains higher spatial resolution than EEG and generates results that are similar to more impractical blood-flow related imaging techniques such as *functional magnetic resonance imaging* (fMRI).

With today's BCI systems, users must learn how to manipulate their brain activity effectively for a given application, either through operant conditioning or by executing certain predetermined cognitive tasks that are distinguishable to the sensors. Imagining the performance of a motor skill, for example, exercises specific parts of the brain (Curran & Stokes, 2003). This specific activity may be detectable by an imaging technique of coarse resolution. Alternatively, applications can be specifically designed to take advantage of naturally occurring brain activity, such as that associated with a flashing light. Under carefully controlled conditions, it is possible to classify the user's engagement in cognitive tasks, such as rest, mental arithmetic, and mental rotation (Kiern & Aunon, 1990).

SIGNAL PROCESSING

It is rare to find a sensor precisely suited to a given sensing task. Often, sensor output must be manipulated or combined with other sensors to fit the needs of the application. This section surveys *signal processing* techniques useful in applying sensors to input and recognition tasks.

Preprocessing

Preprocessing refers to the earliest stage of processing sensor signals. It is at this stage that noise may be removed from raw sensor signals, or signals may be reduced to make them more compact and otherwise easier to use in later processing stages.

The performance of a sensor relevant to preprocessing can be characterized in several ways. *Accuracy* refers to the degree to which the sensor readings represent the true value of what is measured. *Precision*, or *resolution*, by contrast, refers to the extent to which successive readings of the same physical phenomenon agree in value. It is important to realize that while a device's resolution is often measured in bits, this number is often distinct from the number of bits used to represent or store the sensor's readings.

In general, accuracy and precision can be estimated by collecting several successive measurements (*samples* or *observations*) of the same input, and computing the resultant mean and scatter (covariance). An accurate sensor will put the mean near the true value, and a precise sensor will have a small amount of scatter about the mean. An accurate but *noisy* sensor will have low precision (high scatter), but can still be useful by the Central Limit Theorem from statistics: If we make some assumptions about the noise, and average a sufficient number of successive values we will derive a good estimate of the true value (Hoel, Port, & Stone, 1971).

Averaging of successive sensors readings is but one simple way to *smooth* noisy data to obtain a noise-free estimate of what is measured. Of course, the input in a real application is likely to be changing over time, and the manner in which this average is computed can vary. For example, the *boxcar filter* is simply the average of the last n samples, and is thus easy to implement; however, the boxcar filter suffers because it requires a buffer of samples over which the average is computed, and the resulting estimate will *lag* the true value in the case of a changing signal. Related to the boxcar filter is a technique whereby the estimate is obtained as a weighted average of new observation and previous estimate. This filter is even easier to implement and requires no buffer of previous samples. In this technique, however, each estimate depends on previous estimates as the Poisson distribution over time, such that a very quickly moving signal or a signal with many outliers will result in erratic changes in the smoothed signal.

The *Kalman filter* is a popular technique for filtering time-varying signals and can be used to both smooth and predict a signal. It is the optimal linear filter in the sense that it minimizes the difference between the estimate and the true value, assuming a linear model of the input's changing signal and Gaussian noise (Welch & Bishop, 2004). The most common Kalman filter

for estimating the position of a moving object models the object's state as a linear function of both velocity and the position estimate in the previous time step. The Kalman filter models uncertainty in two ways. First, there is some uncertainty in the linear model (how much do we believe that the linear model is correct?). Secondly, there is uncertainty resulting from instantaneous noise that corrupts the observation (how precise is the sensor?). A properly tuned Kalman filter balances these uncertainties appropriately and suffers fewer problems with lag than, for example, a boxcar filter. When the changing signal is actually linear, it can completely defeat lag due to filtering. An improperly tuned Kalman filter, however, can impart unnecessary lag and *overshoot*, such that the estimate runs past the input before correcting itself.

Often, rather than obtaining a continuous estimate of a sensed quantity, we are interested only in obtaining a binary result: Is the switch on or off? When a user throws on a switch in a real system, the output state of the switch can change very rapidly from off to on and back several times before settling to a single stable state. *Debouncing* techniques combat this effect; one simple technique is to ignore the switch for some small, fixed time after seeing the first change in switch state (e.g., 40 millseconds).

More difficult is the situation in which a truly continuous quantity must be transformed into a binary signal. This is commonly done by choosing a *threshold* below which we report the output as "zero," and otherwise, "one." For example, in using a continuous-valued tilt sensor such as an accelerometer to determine whether a tablet PC is being used in "portrait" or "landscape" mode, it is necessary to transform the tilt information into a binary quantity indicating "portrait" or "landscape." An alternative to a single threshold is a scheme with two thresholds and a region between (a *deadband)* in which no change to the output is made. Similar to debouncing, this approach can prevent fluctuation of the output around a single threshold.

Choosing threshold values is generally challenging, and poorly designed complex systems are frequently awash in thresholds that require modification to achieve acceptable performance. Ultimately, the thresholding process destroys information; depending on the nature of subsequent processing, this loss can be detrimental to a system's overall performance. This is particularly true for borderline cases where a system is likely to make erroneous decisions as the result of either an improperly chosen threshold or a noisy input. Such concerns may be eased by adopting a "soft" threshold that reports intermediate results around the "hard" threshold; the *logistic function* (Bishop, 1995) can be useful in this approach.

The signal's effective range of output, or *dynamic range*, must be often be considered both in thresholding and subsequent processing. The relevant range of the property to be sensed must of course lie within the dynamic range of the sensor. If the dynamic range changes—as a consequence, for example, of temperature change, lighting change, or even variations in installation—it may be necessary to calibrate the signal to achieve a normative range. One strategy is to find the sensor's minimum and maximum output during normal use, and to map these to some canonical range. For example, the output of a photodetector may be mapped to a range from zero to one by recording the value of the sensor in the darkest and brightest conditions of regular use. Another strategy is to calibrate the sensor to *ground truth* values that are collected by some other more trusted sensor. Both of these approaches require care if the sensor is not linear in its response; it may then be necessary to fit a curve (e.g., polynomial) to map the sensor to normal values. Characteristics such as dynamic range, linearity of the response, and variation due to temperature are often detailed in a sensor's "data sheet," available from the manufacturer.

In time-varying systems, we are often concerned with the frequency with which we receive new samples from the sensor. An overly high *sampling rate* can result in too much data to process, and can be reduced by *downsampling*. By contrast, many interactive systems will seem to lose their responsiveness if the overall latency is greater than 100 milliseconds. *Latency* or *lag* refers to any delay present in the sensor's response to a change in the sensed property of the world, and can limit the responsiveness of an interactive system built on the sensor (MacKenize & Ware, 1993). A low sampling rate imparts latency, which may be remedied by predictive techniques such as the Kalman filter.

Finally, it is important to consider the true distribution of any noise in filtering and many subsequent processing techniques. Many techniques—including the simplest averaging, Kalman filters, and many probabilistic approaches—assume a Gaussian or uniform distribution of noise. Outliers violating this assumption can be troublesome and should be removed by ad hoc means or techniques from the field of *robust statistics* (Fischler & Bolles, 1981; Huber, 1981). For example, the *median filter*, in which values of a sequence are replaced by the median value, is easy to implement, yet more robust than simple averaging.

Feature Selection

In the context of recognition, a *feature* can refer to a particular sensor or a piece of information derived from one or more sensors, or even derived from other features. Often thought of as a preprocessing step, *feature selection* refers to the process of determining which features are to be computed from the raw inputs and passed to the next level of processing. Appropriate feature selection can sometimes make difficult recognition problems easy. For example, one somewhat unusual approach to detecting faces in video is to detect eye-blinking patterns. Blinking provides a signal that is easily detected by simple image processing operations, and is further supported by the fact that both eyes blink together and are arranged in a symmetric spatial configuration on the face. Blinking thus may be highly *diagnostic* for faces (Crowley & Berard, 1997).

Feature selection begins by determining a set of sensors relevant to the task at hand often with knowledge of the task or domain. In the course of development of a new sensing-based system, it can be beneficial to incorporate as many physical sensors as possible, with the idea that subsequent feature selection processes will indicate which sensors are necessary and sufficient. Furthermore, a number of sensors taken in combination may provide the best overall performance.

Having selected a number of sensors, often the next step is to compute derived features from the raw sensor inputs. For example, when an unimportant and unpredictable offset is pre-

sent in the sensor inputs raw levels, it may be easier to work with its derivative instead. Again, these features are determined in an ad hoc fashion, in light of special domain or application knowledge. For example, early stylus gesture recognition algorithms relied on simple derived features such as the (a) initial angle of the stroke, (b) maximum speed obtained, (c) size of the containing bounding box, (d) duration of the stroke, (e) amount of change in curvature along the gesture, and so on (Rubine, 1991). Early face recognition approaches relied on features such as the distances between facial features such as the eyes, nose, and mouth (Zhao, Chellappa, Phillips, & Rosenfeld, 2003). In the domain of audio, the Linear Predictive Coding (LPC) and the Fourier transform are useful derived feature spaces. The Fourier transform in particular has general applicability to signals with periodicity. For example, the Fourier transform of a body-worn accelerometer may reveal patterns of the user's walking (Hinckley et al., 2000).

Often, it is desirable to put spatial features in a local coordinate system. For example, a gesture recognition system may begin with the position of the head and hands of the user. To remove the effects of the person moving about the room, the head position may be subtracted from the position of each hand, yielding a "head-centric" coordinate system. In the spirit of asymmetric bimanual models of gesture, we might also consider a coordinate system centered on the nondominant hand (Guiard, 1987; Hinckley et al., 1998). In many cases switching to a local coordinate system eliminates a large source of the irrelevant variation present in the raw signal, thus easing subsequent modeling, and can be superior to using only derivative information.

If there is a large number of sensors, or if each sensor is of high dimension (e.g., images taken from video cameras), each sensor's value is unlikely to be statistically independent from one another. To remove redundancy and make the input a more manageable size, some form of *dimensionality reduction* may be used to transform each observation into one of lower dimension. One broad class of techniques involves approximating each sample as the linear combination of a small number of basis functions; the coefficients in this linear combination form the corresponding sample in the new, smaller feature space. *Principle Components Analysis* (PCA) is a popular technique and is the optimal linear technique in the mean-square error sense. PCA finds orthogonal vectors ("principle components," or *eigenvectors*) in the input space as basis vectors, each vector reducing variance (scatter) in the data set. PCA has been used in a wide variety of recognition systems, such as face recognition from images, where often less than 50 components are necessary to perform recognition (Pentland, Moghaddam, & Starner, 1994). Today, there are numerous techniques related to PCA, many of which are more suited to classification (Fodor, 2002).

Where the number of input features is not large, automatic feature selection techniques may be used to determine the subset of features that matter. While the topic is an active area of research, one technique of general applicability is *cross validation* (Bishop, 1995; Mitchell, 1997). The simplest form of cross validation is the *boldout method,* which begins by dividing the data set into two halves. Several variations of the model are then trained on one half of the data and tested on the other. The variation with the best performance on the test set is selected as the best model. In the case of feature selection, each variation employs a particular subset of the original input features; after trying all such subsets, we are left with the best performing subset of features. For more than a handful of original features this approach will be impractical, so various greedy approximations are often used, such as starting with the full set and eliminating one at a time, or successively adding features from a small set.

Classification and Modeling

Classification refers to the process of determining which of several known classes a given sample or observation is drawn from, and is typically the means by which a novel input is *recognized*. A classifier can be used, for example, to recognize which of several known gestures the user has performed by the motion of the pen on a tablet. *Detection* refers to determining the presence of an observation drawn from a known class against a background of many other observations. The distinction between classification and detection is often rather semantic. For example, a face detection system will determine if there is any face present in an image, while a face recognition system will determine the identity of the detected face. While both operations can be thought of as classification, often they call for different techniques.

When simple thresholding or feature selection operations are not enough to transform a group of sensor readings into a signal that is readily consumed by the application, it is often necessary to exploit more sophisticated classification and modeling techniques. These techniques are particularly useful in cases where it is necessary to use many sensors together, and when there are dependencies among them that are difficult to untangle by simple inspection. *Modeling* refers to the choices in representation of sensor values, their dependencies, and the computations performed on them.

There are many ways to classify a new sensor observation as belonging to one of several known classes. Approaches in which a model is trained automatically from a set of training examples are the most relevant to sensor-based systems. These techniques are typically the domain of *machine learning*. The canonical introductory technique is *Fisher's linear discriminant* (Bishop, 1995) in which a closed-form training procedure determines a line in the feature space that optimally divides two classes of training data. A new, unseen sample may then be classified by determining which side of the line the sample lies. Beyond the two-class case, samples are often classified by computing the *likelihood* that the sample was drawn from each class, and choosing the class with the largest likelihood. Assuming a new observation $\mathbf{x}$, and classes C_i we choose C^* as the maximum value $P(C_i)P(\mathbf{x}|C_i)$. The *prior* $P(C_i)$ indicates our belief that a sample is drawn from a class before we even record it, and is often ignored. There are a variety of techniques to derive such probabilistic models from a set of examples.

Common to all these approaches is the ability to characterize the quality of a recognition result. A sample that is correctly classified as belonging to a given class is a *true positive.* A sample that is incorrectly classified as belonging to the class is a *false positive.* A sample that is correctly classified as *not* belonging to a given class is *true negative,* while a sample that is incor-

rectly classified as *not* belonging is a *false negative*. In the context of interactive systems, a false negative might correspond to when the user provides an input and the system fails to recognize it. A high false-negative rate can lead to an overall impression of unresponsiveness, or a sense on the part of the user that they are doing something wrong. False positives, on the other hand, may correspond to when the system takes an action when the user had no such intention, and can lead to an impression that the system is erratic or overly sensitive (Zhai & Bellotti, 2005).

In most situations, a clear trade-off exists between the rate of true positives and false positives. Lower the bar for acceptance to increase the true positive rate, and the rate of false positives is likely to increase. In the context of interactive systems, this tradeoff is especially important to consider when developing criteria for when the system takes action as the result of a recognition process. The *receiver operator characteristic curve* (ROC curve) plots true positive rate against false positive rate and best characterizes this trade-off (see Fig. 10.1). The ROC curve is also an established method to compare the performance of recognition techniques, without regard to any application-specific choice on how tolerant we are to false positives.

In a given application, it is also instructive to break out classification performance by each class. The *confusion matrix* summarizes how a labeled test set is classified by each class, and may reveal that much of the overall classification error can be traced to errors classifying observations from a small number of classes. This can thus inform design of the classifier or the set of application-relevant categories. *Boosting* is one technique in which misclassified samples are emphasized in subsequent training of the model to reduce the overall error rate (Schapire, 2003).

The *naïve Bayes* classifier assumes that the value of a given feature is independent of all the others. This property of *conditional independence* may not actually apply to the data set,

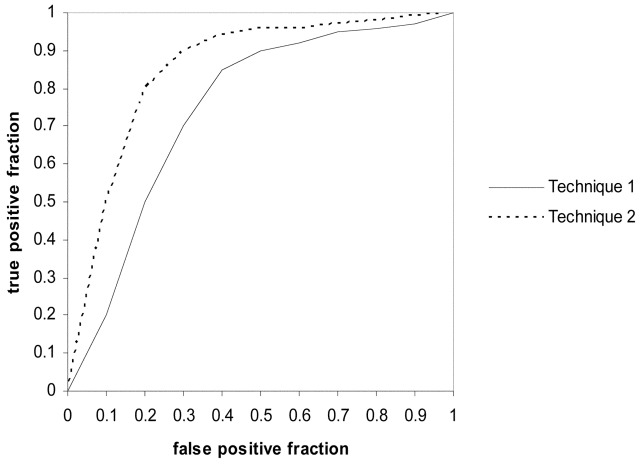


FIGURE 10.1. ROC curves illustrate the trade-off between the rate of true positives and false positives, and can be useful in comparing recognition techniques. Here we see that for a given tolerable rate of false positives, Technique 2 yields better recognition performance than Technique 1.

but its assumption simplifies computation and often may not matter in practice (hence the label "naïve"). Assuming observations of the form $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$, the *posterior probability* of a class C is $P(C|\mathbf{x}) = P(C)\ \mathrm{P}(\mathbf{x}|C)$ by the Bayes rule. Naïve Bayes treats each feature as independent: $P(C|\mathbf{x}) = P(C)\prod_i P(x_i|C)$. Because each feature is modeled independently, naïve Bayes is particularly suited to high dimensional feature spaces and large data sets. Each feature can be continuous or discrete. Discrete variables are often modeled as a histogram (or *probability mass function*), while continuous variables can be *quantized* or *binned* to discrete values, or modeled as a Gaussian or other parametric distribution.

A number of other popular classification techniques do not have obvious probabilistic interpretations. The *neural network*, for example, is best thought of as a function approximation technique. Often, as applied to classification, the input of the approximated function is the observation itself, and the output is a vector whose ith component indicates belief that the observation belongs to the ith class.

Decision trees can be a powerful classification technique that leads to very compact representations for some problems. Each node of a decision tree corresponds to an assertion about the value of a feature in the observation, and yields a split in the data set. The leaves of the tree then indicate the class to which the observation belongs. Classification of a new sample is then rather like the children's game of "twenty questions." Training the model involves determining how to make the splits to optimize classification performance, and possibly, the size of the tree (Breiman, Freidman, Olsen, & Stone, 1984).

The *Support Vector Machine* (SVM) is a powerful modern alternative to the Fisher linear discriminant (Cristianini & Shawe-Taylor, 2000). SVMs determine the split between the two classes to maximize performance on unseen data. Furthermore, SVMs gain much of their power by allowing nonlinear splits of the feature space, but are often thought of as being computational intensive (though, see (Platt, 1999)).

Where the conditional independence assumption of naïve Bayes is too strong, other techniques that directly model the joint probabilities are applicable. For example, a *mixture of Gaussians* uses a sum of multiple Gaussian distributions to model arbitrarily complex joint distributions: $\mathrm{P}(\mathbf{x}|\mathrm{C}) = \sum_i \mathrm{P}(\omega_i)\ \mathrm{P}(\mathbf{x}|\omega_i)$, where $\mathrm{P}(\mathbf{x}|\omega_i)$ is Gaussian with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$. Such mixture models may be trained by the *expectation maximization* (EM) algorithm (Mitchell, 1997; Neal & Hinton, 1999). The EM algorithm is very similar to clustering approaches such as *k-means*, in which k points in the feature space are chosen as representative of the overall set of samples.

Often, there are advantages in treating some subset of the variables as conditionally independent from others. For example, a full joint probability distribution can require a lot of data to train; there may be clear constraints from the application that imply conditional independence, and there may be some subset of the variables that are most effectively modeled with one technique while the rest are best modeled with another. In this case, it may be helpful to selectively apply conditional independence to break the problem into smaller pieces. For example, we might take $P(\mathbf{x}|C) = P(x_1,x_2|C)\mathrm{P}(x_3|C)$ for a three-dimensional feature space, model $P(x_1,x_2|C)$ with a mixture of Gaussians, and $P(x_3|C)$ as a histogram. This overall model amounts to an

assertion of condition independence between x_3 and the joint space of x_1 and x_2.

This modularity afforded by assumptions of conditional independence is taken to its logical conclusion in the *Bayesian network,* which is commonly represented as a directed acyclic graph, where each node corresponds to a random variable x_1 with probability distribution $P(x_i|parents(x_i))$, and each variable is conditionally independent of all variables except its parents (Jensen, 2001). Nodes in a Bayesian network for which we have observations are called *evidence* nodes, while others are considered *hidden.* Observations may be entered into network, and through an *inference* procedure, the likelihood of the observation may be calculated, as well as posterior distributions over any hidden nodes. With Bayesian networks, designers may craft complex probability models without becoming mired in mathematical notation, and software packages allow graphical manipulation of the networks directly (Kadie, Hovel, & Horvitz, 2001).

The *dynamic Bayesian network* (DBN) models time-varying sequences, and thus is relevant to systems in which interactions take place over durations of time. A DBN can be thought of as a Bayesian network where certain nodes depend on the same Bayesian network instantiated on the previous time slice. Dependencies can be within a time slice, or across the time slice. For example, a state variable may depend on its past value as $P(x_t|x_{t-1})$. Such relationships with past values of the same variable can encode a probabilistic finite state machine or *Markov model*, where the distribution $P(x_t|x_{t-1})$ is considered a transition matrix. With this dependency on random variables in the past DBNs can effectively encode a time-varying state or "memory" that may be relevant for an interactive application. For example, it can be useful in modeling interactions composed of a sequence of steps, or where application state itself can be modeled as a finite state machine. Finally, by making a strong dependency on the immediate past, the model can be given some inertia or "smoothed."

One popular special case of the DBN is the *Hidden Markov Model* (HMM), often used to model time-varying signals such as speech, gesture, pen strokes, and so on. HMMs model observations y_t conditioned on a state variable x_t, which evolves over time as $P(x_t|x_{t-1})$. As with many probabilistic models, HMMs are *generative* in nature, meaning one of the ways we can understand them is to consider "running them forward" to generate new observations: an HMM can be thought of as a stochastic finite state machine (Markov model) that generates observations drawn from a distribution associated with each state. With each time step, the Markov model takes a transition to a new state. In the inference (recognition) process, the posterior distribution over the hidden state variable x_t is computed from the observation sequence y_t (Rabiner, 1989). HMMs have been applied many types of observation sequences, including hand gestures, handwriting recognition, speech recognition, and so on. In the simplest application paradigm, a separate HMM is trained for each class.

Much of the attraction of Bayesian networks is due to their flexibility to implement complex probabilistic dependencies. Many probability models may be thought of as particular Bayesian networks. Naïve Bayes, mixture of Gaussians, Hidden Markov Models, and Kalman filters, are among the models that have been shown to be special cases of the Bayesian network (Jordan, 1999). The structure of the Bayesian network is often determined by hand using application-specific knowledge, while the various distributions may be tuned by training from data. Parts of a Bayesian network may be completely handcrafted, derived from domain knowledge in the same manner as expert systems. Other sections of the network may in fact be Hidden Markov Models, Kalman filters, mixtures of Gaussians and various hybrids, such as a mixture of Kalman filters. The automatic learning of the structure of the network itself is active area of research (Heckerman, Geiger, & Chickering, 1994).

Many of the techniques outlined above can be applied to the more generic task of *modeling*, where we are interested in more than classification results. For example, a Bayesian network that fully models the user's interactions with a mobile device might include a variable representing the user's location. The value of this variable will be hidden (unknown) if there is no sensor to directly observe the user's location. We may, however, compute the *posterior distribution* of the variable after several other kinds of observations are entered in the network and find that the user's location is sometimes known with some precision (e.g., the device recognizes the nearest wireless access point with a known location). Not only is this model useful in deducing the user's location, it also enables other parts of the model to exploit this knowledge even if we are not ultimately interested in location information.

Finally, sometimes the signal processing task for an application is better thought of as approximating a function that directly maps a set of inputs to outputs. Techniques such as neural networks, *Radial Basis Function networks*, and other *manifold learning* techniques can be useful in learning mappings from sensor inputs to application-specific quantities. Such techniques can be particularly useful in transforming raw, high-dimensional, nonlinear sensor readings into simple calibrated outputs useful in an application. For example, in carefully controlled circumstances, it is possible to map images of a face to gaze angle by providing a number of face image and gaze-angle pairs. A function approximation approach can interpolate over these examples to map new images to gaze angle (Beymer & Poggio, 1996).

AN EXAMPLE SYSTEM

The following example demonstrates a number of the techniques described above in a working, interactive, sensor-based system. After motivating the overall design of the system, a number of aspects of hardware design are illustrated. Also described are subsequent signal processing steps such as sensor fusion, the application of Bayesian networks for modeling, and gesture and speech recognition.

In the design of intelligent rooms, the issue of how the room's inhabitants might best interact with the room often arises. The traditional notions of desktop computing or the multiple, incompatible, button-laden remote controls typical of consumer electronics are perhaps antithetical to the seamless and untethered experience that is a main feature of the vision of intelligent environments. One popular notion of how users could control an intelligent room is borrowed directly from Star Trek:

The user of the room merely speaks to it, as in, "Computer, turn on the lights."

In the development of one intelligent room (Brumitt, Meyers, Krumm, Kern, & Shafer, 2000), a user study was conducted to determine how real users might want to control multiple lights throughout the space (Brumitt & Cadiz, 2001). A Wizard of Oz paradigm was adopted so that the study would not be limited to designs already implemented. The experimenter, seated behind one-way mirrored glass, operated the lighting controls manually in response to user actions. The users were then exposed to multiple ways of controlling the lights: (a) a traditional GUI list box, (b) a graphical touch screen display depicting a plan view of the room with lights, (c) two speech only-based systems, and (d) a speech and gesture-based system. The study concluded that, like Captain Kirk, users preferred to use speech to control the lights, but that the vocabulary used to indicate which light to control was highly unpredictable. This variance in speech chosen poses a problem for the pure speech-based interface.

Interestingly, the majority of subjects looked at the light they were trying to control while speaking. This observation suggests that an intelligent room could resolve ambiguity in spoken commands (e.g., which light to control) by using computer vision techniques to determine the user's gaze, at least where the device under control is within sight. There are a number of general approaches to computing gaze, but each has serious drawbacks. For example, it is possible to roughly compute gaze from a small number of cameras throughout the room (Wu et al., 2000), but such systems presently lack accuracy and reliability, or require a large number of cameras to cover a useful space. Wearing a special device such as glasses solves some problems, but may not be acceptable to casual users. Another approach is to embed a camera in the device to determine whether the user is looking at it, rather than computing general gaze (Shell et al., 2004). This technique can be effective, but presently scales poorly to a large number of devices.

In light of the difficulties of determining gaze reliably, we reasoned that pointing gestures may play a similar role as gaze in indicating objects. While few subjects in the lighting study spontaneously used gestures, this may be partially explained by the near perfect performance of the Wizard of Oz speech recognizer (the experimenter). Furthermore, pointing may have certain advantages over gaze. For example, pointing is typically the result of a conscious decision to take action, while changes in eye gaze direction may be more involuntary (Zhai et al., 1999). On the other hand, pointing may be no easier to detect by computer vision techniques than gaze (Jojic, Brumitt, Meyers, Harris, & Huang, 2000).

To demonstrate the utility of the combination of pointing and speech as an interface modality in an intelligent environment, we built a hardware device to sense pointing gestures and developed associated signal processing algorithms to combine speech and gesture (Wilson & Shafer, 2003). At the center of the XWand system is a handheld device that may be used to select objects in the room by pointing, and a speech recognition system for a simple command and control grammar. To turn on a light in the room, the user may point the wand at a light and say, "Turn on." Because the pointing gesture serves to limit the context of the interaction (the light), the speech recognition task is reduced to recognizing the few operations available on lights: "Turn on" or "Turn off." Alternatively, the user may perform a simple gesture in place of speech to effect the same command. The user may, for example, point at a media player device, hold the button down, and roll the device to adjust the volume. The XWand system illustrates a number of points related to sensor and recognition-based input, including the hardware design of a composite inertial sensor, sensor fusion, dynamic Bayesian networks, and a host of design considerations.

The original XWand system is based on a 3D model of a room and the controllable devices within it. Using onboard sensors, the XWand device can determine its own absolute orientation, while a computer vision system mounted in the environment finds the position of the wand. Given the size and 3D position of an object in the room, it is a simple trigonometric calculation to determine whether the XWand is currently pointing at the object.

The original XWand hardware device contains an Analog Devices ADXL202 two-axis MEMS accelerometer, a Honeywell HMC1023 three-axis magnetometer, a Murata ENC-03 one-axis piezoelectric gyroscope, a 418MHz FM transceiver, a PIC 16F873 microcontroller, an IR-LED, and a pushbutton mounted on a custom printed circuit board (PCB) (see Fig. 10.2). While the accelerometer is useful in detecting pitch and roll (recall that gravity is an acceleration), it cannot detect the yaw attitude of the device. The three-axis magnetometer reports direction cosines against magnetic north, from which yaw can be determined only if the device is held flat (some GPS devices are equipped with two-axis magnetometers that give heading when the device is held flat). Fortunately, pitch and roll information from the accelerometers may be used to "correct" the output of the three-axis magnetometer to yield a full 3D orientation with respect to magnetic north.

To compute the 3D position of wand, the XWand system uses a pair of FireWire cameras mounted in the corners of the room, which are used to track the IR-LED on the device. Each camera uses an IR-pass filter so that in a typical office environment, only the IR-LED is visible in the image. Furthermore, the IR-LED is programmed to flash at 15Hz. When the host takes the video output of each camera at 30Hz, consecutive images may be subtracted pixelwise so that only objects blinking at 15Hz remain. The IR-LED can thus be located easily in both views. Furthermore, the cameras are calibrated to the geometry of the room so that the 3D position of the IR-LED is obtained from its 2D position in both views. Note that this arrangement assumes a line of sight to the IR-LED from both cameras.

To support speech recognition, an open microphone (low impedance) is placed in the environment. Ultimately, this microphone should be placed on the device, perhaps with the audio encoded and relayed off-board for recognition. The speech recognition engine is programmed with simple command and control grammar based on a simple *command-referent* pattern, where a *referent* can be a device in the environment (e.g., a light) and the *command* refers to one of a number of permitted actions on the device (e.g., "turn on").

Simple gestures made with the wand—such as flicking left, right, up, down, and roll–are recognized by simple routines that measure the change in attitude of the wand from the attitude

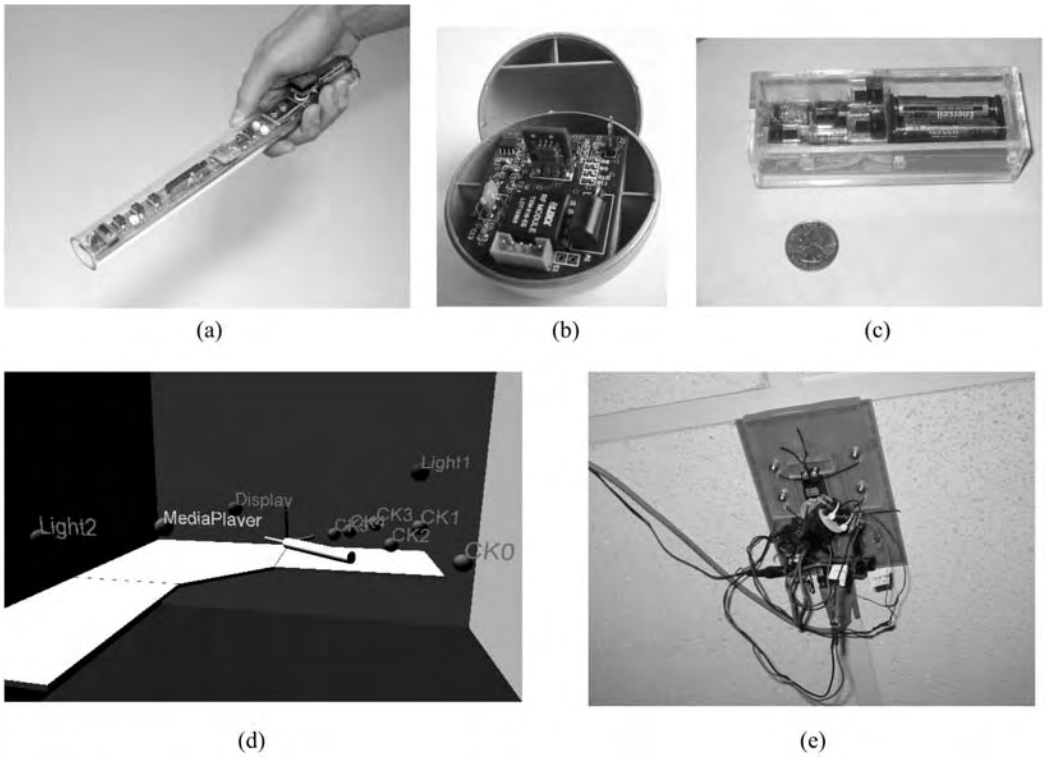


FIGURE 10.2. (a) The first XWand prototype includes accelerometers, magnetometers, gyro, radio, etc. (image © ACM, 2003) (b) The Orb device repackaged much of the XWand. (c) The WarpPointer updates most of the components and uses Bluetooth. (d) The XWand 3D geometry model includes the 3D position of all interactive devices in a room. (image © ACM, 2003) (e) The WorldCursor teleoperated laser pointer is driven by the XWand.

recorded when the button on the device is first pressed. We have also experimented with using Hidden Markov Models to recognize more complex gestures, but have chosen instead to exploit a small set of simple, memorable, and reusable gestures in conjunction with other contextual information such as pointing and possibly speech information. This approach allows for a very simple and robust gesture recognition process, and avoids training users on a gesture set of greater complexity.

A dynamic Bayesian network fuses the various quantities to arrive at a multimodal interpretation of the user's interaction. It models the combination of the output of the speech recognition, the object at which the wand is currently pointing, any gesture performed, the known state of the devices under control, and the state of the interpretation in the previous time steps (see Fig. 10.3). The network bases this combination on the *command-referent* pattern outlined above, where the referent may be determined by speech or pointing gesture, and the command may be determined by speech, gesture, button click, or any combination thereof. The ultimate action to be taken (e.g., "turn on light #2") depends on the command and referent, as well as the state of the device itself (e.g., "turn on light #2" is only permitted if the light is off). Finally, both the command and referent at the current time step depend heavily on the com-

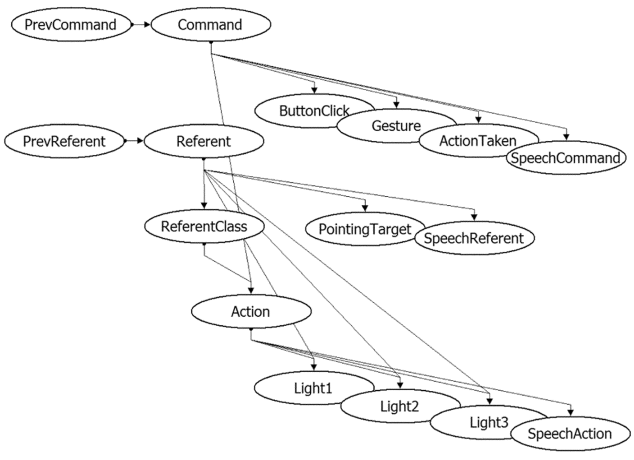


FIGURE 10.3. The XWand Dynamic Bayesian Network (DBN) models multimodal interaction. It combines wand input (Pointing Target, Gesture, ButtonClick), speech input (SpeechReferent, SpeechCommand, SpeechAction), and world state (Light1, Light2, Light3) to determine the next action (Action) as a combination of command (Command) and referent (Referent) and past beliefs (PrevCommand, PrevReferent). (image © ACM, 2003)

mand and referent from the previous time step, such that either quantity can be specified at slightly different moments in time.

The multimodal fusion Bayes network offers a number of interesting capabilities for the XWand system. For example, when the state of each controlled device is represented in the network itself, distributions over related quantities change in appropriate ways. For example, if the user points at a device that is currently turned off, speech and gesture recognition results inconsistent with that state are ignored, such that the phrase "turn off" is removed from the speech recognition grammar. In future work, it may also be possible to infer vocabulary by training the network dynamically (e.g., point at a light and label it as a "light," or point several times at a point in space to train the position of a known light while saying, "Light."

We have continued to develop the XWand prototype in various ways. A typical objection to the camera-based tracking system is the lengthy setup and calibration procedures such systems require. We therefore explored an alternative configuration that eliminates cameras in favor of a single teleoperated laser pointer mounted in the ceiling (Wilson & Pham, 2003). The WorldCursor laser (see Fig. 10.2) is programmed to match the motion of the wand, in a manner similar to how the standard mouse controls a cursor with relative motion. Because the set of objects the laser can reach is limited by line of sight, the original 3D model of the system is eliminated in favor of a simpler spherical coordinate system with an arbitrary origin, thus simplifying setup.

The most recent iteration includes a three-axis accelerometer that can be combined with a magnetometer to arrive at a true 3D orientation by a simple cross-product calculation (see Fig. 10.2). The device has also been applied to cursor control with very large displays, where the mixture of sensors enables a variety of cursor control mechanisms including absolute, orientation based pointing (position and orientation), relative angular motion similar to the Gyromouse, and pure absolute position only. This flexibility allows exploration of several new modes of interaction. The relative, gyro-based mode of pointing allows very fine control with clutching over a small area With very large (wall-sized) displays, however, it is easy to lose the cursor among the many onscreen objects. With the current device, it is possible to momentarily adopt one of the more absolute pointing modes to "warp" the cursor to a point directly in front of the user.

CONSIDERATIONS IN DESIGNING RECOGNITION-BASED SYSTEMS

The previous section touches on issues common to complex sensing-based interactive systems, from hardware design to modeling and sensor fusion. Overall, the strategy in the example is to avoid complex recognition problems whenever possible through thoughtful choice of hardware sensors, design of the interaction, and strong modeling of context. In the following section, we expand upon the motivations behind these design choices. Many of these considerations must be taken into account in the design of recognition-based interactive systems. While many involve difficult problems with no easy answer, it is best to be aware of these issues when designing a sensing-based interactive system.

Computational Cost, Utility, and the Cost of Failure

A number of challenges exist in designing an interactive system that uses sensors and recognition techniques. First, although Moore's Law continually pushes forward the frontier of practical signal processing techniques, many of these algorithms are still computationally intensive. Only in the last six years or so have real-time computer vision techniques become practical on commodity hardware. Furthermore, many of the machine learning and pattern recognition techniques are data-driven, thus requiring large amounts of storage, memory, and training. In the case of mobile devices, where computational power lags desktop computing power by several years, many of these algorithms are impractical, and developers are often forced to make hard choices and take shortcuts. These concerns are often magnified in interactive systems that entirely remake the user interface and employ multiple recognition-based techniques.

Second, recognition-based systems often face serious challenges achieving and guaranteeing the level of robustness required in real applications. In the case of developing consumer products, for example, it is one thing to demonstrate a technique in a laboratory setting; it is quite another to show that the same technique will work in the variety of circumstances in which customers will expect it to work. Unit testing even simple recognition-based interactive systems can be daunting. Computer vision techniques, for example, are often susceptible to variations in lighting, while audio-based techniques may fail in the presence of background noise. Some effort has been devoted to developing signal processing techniques that adapt to both the current circumstances and user. Adaptive speech recognition and handwriting recognition techniques, for example, have become commonplace. Even in these cases, however, it is important that the system have good functionality out of the box or else the user may not use the system long enough for an adaptive algorithm to improve performance.

In the development of a recognition-based interactive system, it may become impractical to seek more improvement in recognition performance. At this point, it is important to consider the cost of recognition failure: often the cost of repairing a false positive recognition can overwhelm any advantage in the use of the system. In speech recognition, for example, the repair of the errors can be awkward, slow, and disruptive to the task (C.-M. Karat, Halverson, C. Karat, & Horn, 1999). Only users that are unable to use a regular keyboard may accept a dictation system that fails three times out of one hundred words, for example (Feng, C.-M. Karat, & Sears, 2005). Another consideration is whether the system returns a result similar to the desired result when it fails ("graceful degradation"), in which case repair is likely to be easier (Horvitz, 1999). If a recognition failure is too costly to consider repair (for example, control of an air-lock on a spacecraft, or more mundanely, closing a window on a desktop GUI), the cost of making a mistake may be incorporated directly in the model so that false positives are more likely to be avoided. This can be done either by seeking some kind of deliberate confirmation from the user, or more simply by moving

thresholds up the ROC curve. In the latter approach, it is important to be aware that often users modify their behavior on successive attempts such that their input is no longer modeled by the usual training corpus; in speech this is known as the Lombard effect (Junqua, 1993), but this phenomenon can be observed in other modalities as well.

Considerations of computational cost, robustness, and cost of errors should play prominent roles in the end-to-end design of recognition-based systems. Ultimately, the designer may be forced to recast the interaction and possibly the application altogether. An example of a system in which one can find many of these issues played out is the Sony EyeToy, an add-on camera for the Sony Playstation 2. The EyeToy allows the player to interact with a game through gesture and body motion, rather than through the usual game controller. It also appears to use very simple image processing techniques to determine the player's motion. These techniques are computationally cheap and generally quite robust to varying light levels and other factors that are likely to be encountered in a residential application. Furthermore, the EyeToy works with a small number of games written specifically for use with this device. These games take advantage of the strengths of the EyeToy, rather than risk providing a poor emulation of the regular game controller.

The Role of Feedback

Feedback is important in any interactive application, but may be even more so where sensing and recognition are used (Bellotti et al., 2002). We can characterize some kinds of feedback as "tight" or not, where a "tight" feedback loop provides feedback in a frequent, timely, and informative fashion.

Usually, if there is any chance of recognition failure, the system should provide feedback on all recognition results, errors or otherwise. If possible, the system should provide some indication of the nature of any failure so that the user can modify his or her own behavior to meet the system's expectations. If the system can provide some indication of the quality of the interpretation as it is happening, then it may be possible to allow the user to cancel or modify the interaction on the fly, so as to avoid costly errors or cumbersome confirmation routines (Vogel & Balakrishnan, 2004).

The right feedback can often influence the design of the sensing and recognition algorithm itself. For example, because the onscreen cursor is updated so quickly (a tight feedback loop), a naïve user might not ever realize or care that the mouse provides rapid, small, successive bits of relative movement information rather than true position information, which would be much harder to sense. This trick is used in the WorldCursor system to avoid the use of cameras required by the original XWand system.

Considering the EyeToy again, it is interesting to note that the image of the player is often incorporated into the onscreen presentation. By watching themselves onscreen, players are able to interact with onscreen elements without relying on sophisticated, yet more failure prone and computationally intense, hand-tracking algorithms. This feedback also cleverly ensures that the player stays in the camera's field of view; the flow the game does not have to be broken to alert a player who has left the field of view.

Implicit and Explicit Interaction

In the previous discussion regarding feedback and cost of failure, we assume that interaction is structured in such a way that the user takes action and expects a timely response from the system—hat is, the user's actions and the system's responses are *explicit*. Most interactive systems can be characterized in this way.

In contrast to *explicit* interactions, *implicit* interactions are based not on explicit action by the user, but more commonly on users' existing patterns of behavior (Schmidt, 2000). For example, with the frustration-sensing system outlined in the introduction, the state of frustration is not explicitly entered by the user in order to elicit some behavior from the system. Instead, the state arises naturally and perhaps involuntarily, and upon detection, the system should take appropriate action.

Implicit interactions may take place over a long duration, and may not exhibit an obvious pattern of cause and effect. For example, systems that adapt to perceived user preferences, as indicated by the history of user behavior, might eventually make a recommendation to the user, or even take actions so subtle that the user may not notice them. Such systems can be complex in terms of sensing and modeling, and often tend towards automating or refining aspects of the user's original task (Horvitz, Breese, Heckerman, Hovel, & Rommelse, 1998). For example, a smart home may observe its inhabitants' daily patterns of coming and going to determine an optimal schedule to control the thermostat automatically, balancing comfort and economy (Mozer, 2005). One potential difficulty is that, unless the output of the system is designed very carefully, users may feel unable to correct a mistake made by the system or exert more explicit control in the face of exceptional circumstances (e.g., a party). Designers should consider providing functionality that allows the user to query the system for why it took a given action, provide simple mechanisms to redress errors, and finally, revert to a manual control mode.

Implicit interaction systems driven by patterns of ongoing user behavior are often not as critically dependent on sensing and recognition reliability and the nature of feedback. Rather, it is more important that interpretation processes are more correct than incorrect over time; accordingly, modeling techniques that integrate noisy sensor values over time are often appropriate. The automatic thermostat, for example, should probably incorporate many observations of the users' behavior—including presence and manual thermostat control—and model weekends, weekdays, and holidays differently. Sophisticated modeling techniques in hand, designers of such systems have the opportunity to exploit powerful, though sometimes unreliable, sensing techniques.

The Importance of Context

Notions of *context* can play an important role in sensing-based interaction (Dey, Kortuem, Morse, & Schmidt, 2001). "Context" refers to the overall environment the interaction or device finds itself in, rather than the objects obviously and directly relevant to the task at hand. What constitutes context often depends on point of view; after studying the application in detail, factors that

once may have seemed external and tangentially relevant (context) may be central to the model after all.

Note the purposeful vagueness in the definition of "context;" with respect to sensing, "environment" can refer to the actual physical surroundings. The addition of sensors to a system may give it awareness of its environment, and thereby enable interesting "context-dependent" behavior. For example, a mobile device equipped with GPS or other localization technology might bring up web search results corresponding to the nearest points of interest (Abowd et al., 1997; Davies, Ceverst, Mitchell, & Efrat, 2001; Hariharan, Krumm, & Horvitz, 2005). Beyond the physical environment, context can refer to more abstract states, such the user's current activity (e.g., working or not), the history of their interactions, and their preferences. It may also refer to momentary information, such as the action of the nondominant hand, or anaphoric references that provide scope for an ongoing interaction.

Often our activities follow preestablished patterns, either by design or by accident. These patterns can provide strong contextual information for interpreting the user's interactions. For example, entertainment-based scenarios may have a narrative structure that constrains the interaction (Galyean, 1995). The KidsRoom interactive experience, for example, was structured according to a narrative progression involving the fantastical transformation of a children's room (Bobick et al., 1999; Bobick et al.,2000; see Fig. 10.4). This structure in turn guided the selection of recognition at any given moment.

Modeling context explicitly can be a powerful way to solve difficult recognition problems. The more context can be brought to bear, often the easier and more robust the recognition. Although counterintuitive at first, by exploiting context, the combination of multiple sensors with simple signal processing techniques may result in better performance than the use of fewer sensors with more complex signal processing techniques. Recall how in the XWand system, the speech recognition process is constrained by the knowledge of what the user is currently pointing at with the wand, the current state of the device indicated, and so forth. In fact, the interaction can be so constrained that, often, the system only needs some indication that the user said *anything*. Similarly, it suffices to use simple, learnable and reusable gestures when the interaction has been contextualized by pointing, speech, or both. Our frustration detection system may be more robust if it incorporates the knowledge of the application currently in focus. For example, it may be easier to write one frustration detector for office applications and a separate one for use while playing video games, rather than one detector that works in both contexts. In the end, the two detectors may only differ in terms of how some threshold is set.



FIGURE 10.4. The KidsRoom engaged children to participate in an interactive journey. Computer vision and projection technologies were used to transform an ordinary room into a variety of settings including a forest and river. Image © the Massachusetts Institute of Technology, 1999.

The Importance of A P*riori* Knowledge

The development of a model for a sensing, interactive, system can benefit from specific knowledge of the domain in which the interaction is situated. Such *a priori* knowledge can lead to insights as to how to determine meaningful categories from raw sensor data. Higher level rules taken from the domain can then be brought to bear. For example, in the case of a pen-based system that automatically parses and manipulates mathematical equations, the rules of how mathematical expressions are combined can be a powerful constraint that drives the correct interpretation of the sloppiest drawings (LaViola & Zeleznik, 2004). Similarly, knowledge of chemistry can guide the transformation of a sketches of molecules to a full 3D model (Tenneson & Becker, 2005). Providing a kind of context, strong assumptions taken from the domain can limit the applicability of a model but can often dramatically improve performance.

Many domains of human behavior have been categorized and described in terms of detailed taxonomies and ontologies. For example, music, gesture, dance, and spoken language each have detailed ontologies, notation schemes, and so on. It can be beneficial to draw from such knowledge when developing a model, but some aspects of a categorization scheme may not be fully supported by the available sensors. For example, there may be some aspect of the domain not covered by the model, or a single category may "alias" to several distinct classes as perceived by the sensors. A detailed analysis of the sensing system output may lead to insights on the original domain model.

One of the advantages of incorporating bits of domain knowledge representation directly into the model itself is that it becomes more transparent to its designers and users, and thus, more reusable. If there is a problem with the system, the designer may directly inspect semantically relevant quantities from the model. Approaches that do not rely on such informed representations, such as neural networks, are often so difficult to inspect that upon discovery of a problem, it may be easier to retrain the model from scratch than to troubleshoot what the network really learned. A good compromise is to use the more data-driven techniques (such as neural networks or probabilistic modeling techniques) to map the raw input signals onto semantically meaningful mid-level primitives. Such choices in representation can support modularity and explorative research.

It is interesting to note, however, that initial research on a given complex sensing problem often draws heavily from domain knowledge, only to be eclipsed later by more purely data-

driven approaches. Early approaches to speech recognition, for example, transformed the audio signal into a string of symbols representing phoneme categories developed by linguists; today, one is more likely to see approaches in which *subword acoustic unit* categories are trained from audio signals directly in a purely data-driven approach. Early face detection and recognition approaches similarly relied on prescriptive or ad hoc features, while more recent approaches are more purely data-driven (Li & Jain, 2005; Zhao et al., 2003).

Generalize or Specialize?

The incorporation of a priori and context information leads to potentially complex, yet powerful, modeling techniques. One of the drawbacks of adding more and more detail into a model, however, is that the resulting system may be so tailored to a particular domain or set of contextual circumstances that it fails to generalize to new applications. It seems as if engineering best practices, including modularity and device-independence, run counter to models that are optimized for a given situation.

For example, consider the problem of delivering various location-based services to inhabitants of an indoor space. A good, familiar choice for the representation of each person's location might be two-dimensional Cartesian coordinates on a full map of the environment. A person-tracking system is installed in the environment; it is charged with determining where everyone is on the map. This information is then passed onto another part of the system that delivers services to each user based on the knowledge of where everyone is at any given moment.

Such a choice of representation has many desirable features. We can consider a multitude of person-tracking technologies—as long as each reports 2D Cartesian coordinates we can incorporate its output into our map ("device independence"), and we can consider merging the results of multiple such systems in a simple probabilistic framework where we model the location of each person as a distribution over space rather than simple 2D coordinates. Furthermore, we can exploit familiar rules of geometry to calculate interesting properties such as whether a display is within sight of a particular user. Most importantly, we need not concern ourselves with the particular application services that our ultimate system will provide: as long as each service expects only our generic representation we can expect the service to work properly. The system is *general* in its applicability.

Contrast this approach to one in which we have no unifying geometric model or map, but instead, install proximity sensors at each of the devices of interest, each sensor detecting some one as they approach the device. Perhaps we install special sensors on some of the doors to detect when someone walks through (Abowd, Battestini, & O'Connell, 2003), and maybe a sensor on seats to detect when someone sits down (Brumitt et al., 2000). Often, proximity information is enough and requires no calibration step (Krumm & Hinckley, 2004). For our location-based services, we note that in some cases, it is important to know *who* is at a given location, but in many cases, this information is not necessary. We place various simple sensors throughout the environment in locations that we think will provide the information needed to support the specific location-based services we currently have in mind. The solution we come up with is quite *specialized*.

It is unclear which approach is superior. The geometric model-based approach depends heavily on the performance of the person-tracking system. When it fails, it may return no useful position information, even when the given circumstances do not require precise position information, and even while the sensor and the underlying signal processing algorithms may produce relevant intermediate results that could be used at a subsequent level of processing. The assumption of device-independence ignores the fact that different sensing technologies typically have very different failure modes. The resulting application may perform poorly due its design around modeling choices that follow the lowest common denominator. On the other hand, it could work very well and provide the utmost in flexibility if the person-tracking is very reliable.

The specialized approach of course suffers in all the ways that the general approach excels; each installation of the system may require significant innovation on the part of the designer. On the other hand, because it is tailored to the application, the system is more likely to gracefully handle sensing failure modes, and furthermore, is more likely to be less wasteful of resources, both physical and computational. Not surprisingly, the specialized approach is likely to exhibit better performance for a given application than the generalized approach.

The consideration of generalized versus specialized designs is a common engineering problem that is especially relevant in the realm of sensor and recognition-based systems. As our example illustrates, the two approaches may demand completely different sensors, representations, and modeling choices.

Traditional vs. Nontraditional Interfaces

A question that naturally arises in the application of sensing and recognition to interactive systems is whether the design should emulate, augment, or completely replace the interfaces we already have.

It is probably not surprising that so many interactive sensor-based systems emulate the mouse. After all, once this functionality is achieved, the system is now relevant to the vast majority of the world's software. It is interesting to note, however, the degree to which the original development of the GUI hinged on the development of the mouse itself, and how to this day, the mouse is still the favored input device. This suggests that unless the new interactive system offers significant new functionality over the mouse, it will not be adopted, and that instead of forcing the new techniques on today's interfaces, designers should think about what changes in the interface are implied by new sensing systems.

Another approach is to augment or complement today's interfaces. For example, it is relatively simple to add an accelerometer to a mobile device that allows the user to position the cursor or scroll by tilting the device, and it is not hard to imagine how users could easily pick up this interaction (Hinckley et al., 2000).

Still another approach is to completely reinvent the interface. The risks are that the user is required to learn a completely new way of working, the designer is faced with developing the

entire system, and the utility of the system will inevitably be compared more established techniques. Again, unless there is a significant improvement to be had, users are unlikely to adopt the new approach.

Many of these issues are illustrated by recent research into interactive table systems. In its simplest form, an interactive table system might be no more than a large flat-panel touch screen turned horizontal to function as a table surface. This simple change in orientation enables new applications that exploit the collaborative nature of multiple users gathering around a table (Shen, Vernier, Forlines, & Ringel, 2004). Furthermore, sensing techniques have been developed to support multiple points of contact on the surface, and to support bimanual interaction, or the multiple hands of multiple users. Distinct from wall displays, tables are able hold objects, and accordingly, some interactive tables use sensing techniques to recognize objects placed up on them. This ability can be used to support various *tangible user interface* scenarios. For example, a puck placed on the table can enable a feature of the application; the user might rotate the puck to adjust an associated parameter, or it might call up a group of photos (Fitzmaurice et al., 1995; Ullmer, Ishii, & Glas, 1998; see also Ishii, chapter , this volume.).

Clearly, in order to exploit the unique affordances and sensing capabilities of an interactive table, we must be willing to let go of many of today's GUI interfaces. For example, the multiple-touch and multiple-user aspect cannot be supported by the single-focus model of the GUI, and even the assumption that the display has a natural reading orientation (an "up" direction) may no longer be valid (Kruger, Carpendale, Scott, & Greenberg, 2004; Shen, Lesh, & Vernier, 2003; see also Scott, chapter , this volume). Finally, today's GUI model has no role for tangible UIs.

The designers of interactive table systems tend to follow a few guiding principles to innovate without alienating the user. First, there is a desire to leave the overburdened widgets of the modern GUI behind and instead rely on more of a direct manipulation style of interaction. For example, to rotate a virtual object on the screen, it may suffice to place two fingers anywhere on the object and move one finger about the other, much in the same way that one might rotate a piece of paper sitting on a desk. By using the multiple touch capability, we avoid the clumsy widget-heavy and mode-heavy rotation techniques typical of many drawing packages. The addition of pucks and other tangle UIs extends this approach; for example, while the puck is on the table, it may behave as knob (Patten, Ishii, & Pangaro, 2001). Secondly, there is a trend to add widget-based interaction back into the direct manipulation framework, perhaps mainly to address the need to perform a variety of actions on the digital representation of the object. For example, e-mailing, printing, and contrast and color adjustment are just a few things that the user might want to do with their photos; these operations are outside of the scope of direct manipulation. These widget-based interactions can draw upon the advantages a multiple touch table interface provides. For example, putting two fingers down near an object may trigger a scrollbar-like widget in which bringing the two fingers closer or further apart adjusts a parameter (Wu & Balakrishnan, 2003).

Evaluating Novel Interfaces

Determining whether the novel sensing and interaction model actually works is the domain of usability testing. No research into sensor and recognition-based input is complete without an evaluation showing its effectiveness. Unfortunately, the technical difficulty of getting such systems to work at all often leaves little time to quantify performance on a standard task. Furthermore, when the new work is in a preliminary state, it may not be instructive to compare the new technique against one that has had decades to evolve.

Many times, user study subjects are often so impressed by the "magic" of sensor-based systems that the sheer novelty of the interface can skew study results. Subjective surveys are likely to show bias in favor of the novel design (Nielsen, 1994). This is a very difficult problem without an easy solution, particularly in the case of tangible UIs, tabletop interactive systems, and perceptual user interface systems. *Longitudinal studies,* which involve repeated sessions spread out over multiple days, can be used to minimize this effect, but such studies are expensive and time-consuming. As a result, many interactive sensing-based systems come with few convincing quantitative user studies that prove their utility. In the case of a system that uses tracking or detection to select an object, Fitts' Law studies can be an effective technique to compare pointing performance across very different systems (MacKenzie, 1992).

Of course, often the point of the work is not to show a decrease in task completion time, a reduction in errors, or other more conventional metrics from the field of HCI. Rather, the goal may be to highlight completely new ways of conceptualizing the relationship between users and their machines, and to demonstrate that such innovations are technically feasible. There is often more of an emphasis on invention and design than on evaluation. The novelty that can sabotage a user study may even be a desirable effect, compelling users to engaging experiences they may not have otherwise had. Furthermore, traditional evaluation methods seem at odds with the goals of surprising, delighting, and entertaining the user. The field has only recently begun to recognize the need to develop ways to objectively evaluate interactive systems along these dimensions that are basic to the quality of life (Blythe, Overbeeke, Monk, & Wright, 2004; Norman, 2003).

A fantastic example where simple sensing techniques were used to great effect to surprise and delight the user is the PingPongPlus system (Ishii, Wisneski, Orbanes, Chun, & Paradiso, 1999). In this research prototype, the usual ping-pong table was augmented with sound effects and a top-down video projection onto the table surface, and electronics to sense where the ping pong ball hits the surface during game play (see Fig. 10.5). This sensing system used eight microphones mounted under the table, and custom electronics to triangulate where the ball hit the surface. The sound effects and video presentation reacted to each strike of the ball, in ways calculated to amuse the players and augment the game play in dramatic ways. For example, in one mode, the ball produces ripples on the table, while in another, thunderstorm audio and video effects build up as the length of the volley increases.

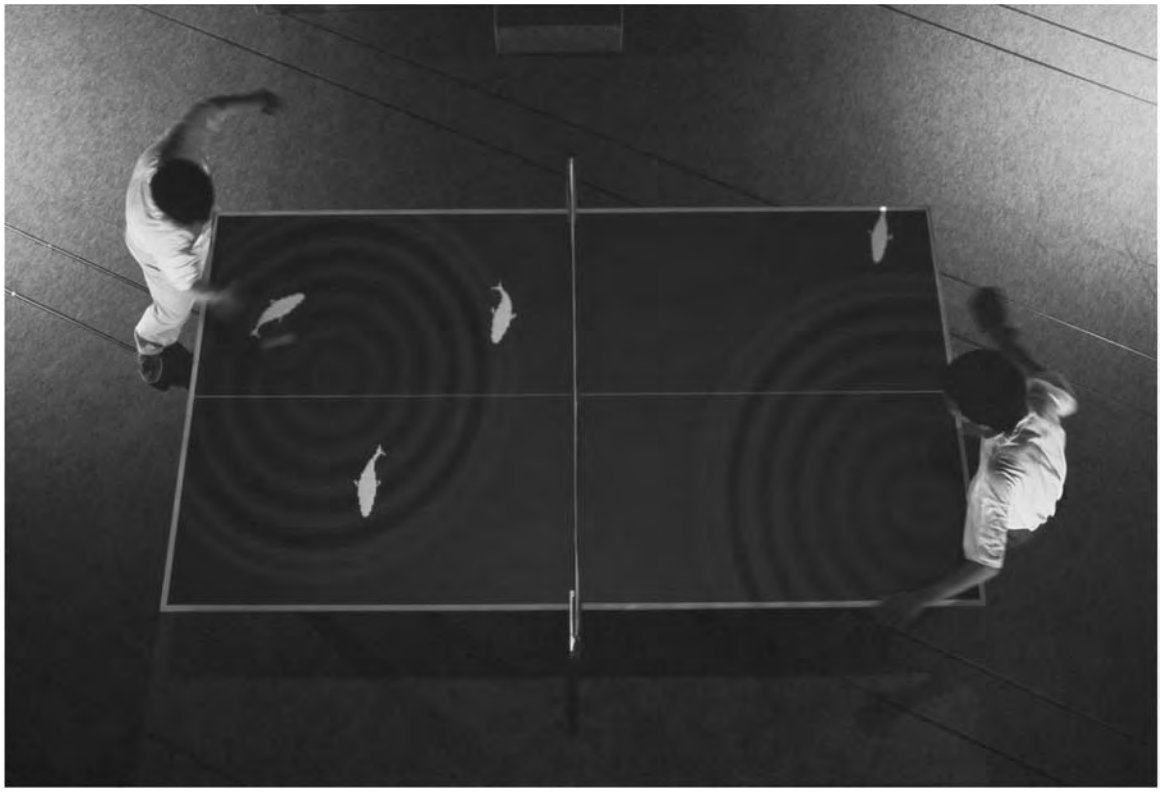


FIGURE 10.5. The PingPongPlus system uses a series of microphones to triangulate where the ping pong ball strikes the surface of the table. This position information is used to drive a variety of interactive graphics displayed on the table by an overhead projector. Image © 2006 Tangible Media Group, MIT Media Laboratory

CONCLUSION

In this chapter, we explored a variety of sensing technologies available today, outlined a number of signal processing techniques common to using these sensing technologies in sensor and recognition-based input for interactive systems, and discussed further issues related to designing such systems.

While much of the discussion highlighted difficulties in designing systems that rely on sensors, the future of interactive sensing systems is bright. Advances in MEMS and nanotechnology will continue to drive innovations in the sensors themselves, while the relentless increase in commodity CPU power and storage capabilities will continue to enable more sophisticated modeling techniques used in interpreting sensor outputs.

Another powerful driver in the development of sensing-based interactive systems is the growth of the computing form factor beyond the traditional desktop computer. The proliferation of cell phones, personal digital assistants, portable gaming devices, music players, tablet PCs, and living room-centric PCs shows a trend towards the vision of *ubiquitous computing,* in which computing is situated throughout our environment and daily life. Individual computing devices will be tailored in deep ways to the task at hand, away from the "one size fits all" mentality of desktop computing. The use of sensors and recognition techniques will play an important role in enabling this diversity, and will naturally support and demand a variety of interaction styles. As interactive systems become more tailored to a given activity, the opportunity to leverage the techniques described in this chapter increases, in turn enabling the application of the sensors themselves. Such a virtuous cycle may speed the development and adoption of sensing-based systems in ways that are hard to imagine today.

ACKNOWLEDGMENTS

The XWand and Orb prototypes were developed with the assistance of Mike Sinclair and Steven Bathiche. Thanks to Ken Hinckley, John Krumm, Steven Bathiche, and Ewa Davison for comments.

References

Abowd, G. A., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks, 3*, 421–433.

Abowd, G. A., Battestini, A., & O'Connell, T. (2003). *The location service: a framework for handling multiple location sensing technologies* (GVU Technical Report GIT-GVU-03-07). Atlanta, Georgia: Georgia Institute of Technology.

Bao, L., & Intille, S. (2004). Activity recognition from user-annotated acceleration data. *Proceedings of the Second International Conference in Pervasive Computing*, 3001/2004, 1–17.

Baudel, T., & Beaudouin-Lafon, M. (1993). Charade: Remote control of objects using free-hand gestures. *Communications of the ACM, 36*(7), 28–35.

Baxter, L. (1996). *Capacitive sensors: Design and applications.* Hoboken, NJ: Wiley-IEEE Press.

Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., & Lopes, C. (2002). Making sense of sensing systems: Five questions for designers and researchers. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 415–422.

Beymer, D., & Flickner, M. (2003). Eye gaze tracking using an active stereo head. *IEEE Conference on Computer Vision and Pattern Recognition*, 451.

Beymer, D., & Poggio, T. (1996). Image representations for visual learning. *Science, 272*, 1905–1909.

Bilmes, J. A., Li, X., Malkin, J., Kilanski, K., Wright, R., Kirchoff, K., et al. (2005). The vocal joystick: A voice-based human-computer interface for individuals with motor impairments. *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 995–1002.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford, UK: Oxford University Press.

Blythe, M. A., Overbeeke, K., Monk, A. F., & Wright, P. C. (Eds.). (2004). *Funology: From usability to enjoyment.* Human-Computer Interaction Series. Dordrecht, The Netherlands: Kluwer.

Bobick, A., Intille, S., Davis, J., Baird, F., Pinhanez, C., Campbell, L., et al. (1999). The KidsRoom. A perceptually-based interactive and immersive story environment. *Pressure Teleoperators and Virtual Environments*, *8*(4), 367–391.

Bobick, A., Intille, S., Davis, J., Baird, F., Pinhanez, C., Campbell, L., et al. (2000). The KidsRoom. *Communications of the ACM, 43*(3), .

Boyd, J., & Little, J. (2005). Biometric gait recognition. In M. Tistarelli, J. Bigun, & E. Grosso (Eds.), *Advanced Studies in Biometrics: Summer School on Biometrics, Alghero, Italy, June 2–6, 2003.* Revised Selected Lectures and Papers, Vol. 3161/2005 (pp. 19–42), New York, NY: Springer.

Brashear, H., Starner, T., Lukowicz, P., & Junker, H. (2003). Using multiple sensors for mobile sign language recognition. *Proceedings of the 5th International Symposium on Wearable Computing,* 45–52.

Breiman, L., Freidman, J. H., Olsen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Boca Raton, Florida: Chapman & Hall/CRC.

Brumitt, B., & Cadiz, J. (2001). Let there be light: Examining interfaces for homes of the future. *Proceedings of Interact'01*, 375–382.

Brumitt, B. L., Meyers, B., Krumm, J., Kern A., & Shafer, S. (2000). EasyLiving: Technologies for intelligent environments. *Proceedings of the Handheld and Ubiquitous Computing, 2nd International Symposium*, 12–27.

Caruso, M. J. (1997). Applications of magnetoresistive sensors in navigation systems. *Sensors and Actuators 1997*, 15–21.

Choudhury, T., & Pentland, A. (2003). Sensing and modeling human networks using the Sociometer. *Proceeding of the International Conference on Wearable Computing*, 216–222.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Crowley, J. L., & Berard, F. (1997). Multi-modal tracking of faces for video communications. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 640–645.

Curran, E., & Stokes, M. J. (2003). Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems. *Brain and Cognition, 51*, 326–336.

Davies, N., Ceverst, K., Mitchell, K., & Efrat, A. (2001). Using and determining location in a context-sensitive tour guide. *IEEE Computer, 34*(8), 35–41.

Dey, A. K., Kortuem, G., Morse, D. R., & Schmidt, A. (Eds.) (2001). Special Issue on Situated Interaction and Context-Aware Computing. *Personal and Ubiquitous Computing, 5*(1).

Dietz, P., & Leigh, D. (2001). DiamondTouch: A multi-user touch technology. *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, 219–226.

Fails, J. A., & Olsen, D. (2002). Light widgets: Interacting in everyday spaces. *Proceedings of the 7th International Conference on Intelligent User Interfaces*, 63–69.

Feng, J., Karat, C.-M., & Sears, A. (2005). How productivity improves in hands-free continuous dictation tasks: lessons learned from a longitudinal study. *Interacting with Computers, 17*(3), 265–289.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*, 381–395.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. (1995). Bricks: Laying the foundations for graspable user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 442–449.

Fodor, I. K. (2002). A survey of dimension reduction techniques. Livermore, CA: Center for Applied Scientific Computing, Lawrence Livermore National Laboratory.

Forsyth, D. A., & Ponce, J. (2002). *Computer vision: A modern approach*. Upper Saddle River, NJ: Prentice Hall.

Fraden, J. (2003). *Handbook of modern sensors: Physics, designs, and applications*. New York, NY: Springer.

Galyean, T. (1995). *Narrative guidance of interactivity*. Unpublished doctoral thesis, Media Laboratory, Massachusetts Institute of Technology, Cambridge.

Garfinkel, S., & Rosenberg, B. (Eds.). (2005). *RFID: Applications, security, and privacy*. Boston, MA: Addison-Wesley.

Greenberg, S., & Fitchett, C. (2001). Phidgets: Easy development of physical interfaces through physical widgets. *Proceedings of the UIST 2001 14th Annual ACM Symposium on User Interface Software*, 209–218.

Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior, 19*, 486–517.

Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, 115–118.

Hariharan, R., Krumm, J., & Horvitz, E. (2005). Web-enhanced GPS. In T. Strang and C. Linhoff-Popien (Eds.), *Location- and Context-Awareness: First international workshop, LoCA 2005, Oberpfaffenhofen, Germany, May 12–13, 2005, Proceedings* (pp. 95–104). New York, NY: Springer.

Haro, A., Flicker, M., & Essa, I. (2000). Detecting and tracking eyes by using their physiological properties, dynamics and appearance. *Proceedings of the IEEE CVPR 2000*, 163–168.

Heckerman, D., Geiger, D., & Chickering, D. M. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. *10th Conference on Uncertainty in Artificial Intelligence*, 293–301.

Hinckley, K., Pausch, R., Proffitt, D., & Kassel, N. (1998). Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI), 5*(3), 260–302.

Hinckley, K., Pierce, J., Sinclair, M., & Horvitz, E. (2000). Sensing techniques for mobile interaction. *Proceedings of the ACM UIST 2000 Symposium on User Interface Software and Technology*, 91–100.

Hinckley, K., & Sinclair, M. (1999). Touch-sensing input devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (SIGGCHI '99)*, 223–230.

Hjelm, S. I., & Browall, C. (2000). Brainball—using brain activity for cool competition. Paper presented at NordiCHI 2000, Stockholm, Sweden.

Hoel, P. G., Port, S. C., & Stone, C. J. (1971). *Introduction to probability theory*. Boston, MA: Houghton Mifflin.

Horn, B. K. P. (1986). *Robot vision.* Cambridge, MA: MIT Press.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 159–166.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumiere project: Baysian user modeling for inferring the goals and needs of software users. *Proceedings of the fourteenth conference on uncertainty in artificial intelligence*, 256–265.

Huber, P. J. (1981). *Robust statistics.* New York: John Wiley & Sons.

Ishii, H., Wisneski, C., Orbanes, J., Chun, B., & Paradiso, J. (1999). PingPongPlus: Design of an athletic-tangible interface for computer-supported cooperative play. *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI)*, 327–328.

Jacob, R. J. K. (1993). Eye-gaze computer interfaces: what you look at is what you get. *IEEE Computer, 26*(7), 65–67.

Jensen, F. V. (2001). *Bayesian networks and decision graphs.* New York, NY: Springer.

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics, 14*, 201–211.

Jojic, N., Brumitt, B., Meyers, B., Harris, S., & Huang, T. (2000). Detection and estimation of pointing gestures in dense disparity maps. *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, 468–475.

Jordan, M. (Ed.). (1999). *Learning in graphical models.* Cambridge, MA: MIT Press.

Junqua, J. C. (1993). The Lombard reflex and its role on human listeners and automatic speech recognizers. *Journal of the Acoustical Society of America, 93*(1), 510–524.

Kadie, C. M., Hovel, D., & Horvitz, E. (2001). *MSBNx: A component-centric toolkit for modeling and inference with Bayesian networks* (Microsoft Research Technical Report MSR-TR-2001-67). Redmond, WA: Microsoft Corporation.

Kahn, J. M., Katz, R. H., & Pister, K. S. J. (2000). Emerging challenges: Mobile networking for 'smart dust.' *Journal of Communications and Networks, 2*(3), 188–196.

Karat, C.-M., Halverson, C., Karat, J., & Horn, D. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. *Proceedings of CHI 99*, 568–575.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., & Tachibana, K. (2000). Virtual object manipulation on a table-top AR environment. *Proceedings of ISAR 2000*, 111–119.

Kiern, Z. A., & Aunon, J. I. (1990). A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering, 37*(12), 1209–1214.

Klein, J., Moon, Y., & Picard, R. W. (2002). This computer responds to user frustration: Theory, design and results. *Interacting with Computers, 14*(2002), 119–140.

Kovacs, G. T. (1998). *Micromachined Transducers Sourcebook.* New York: McGraw-Hill.

Kruger, R., Carpendale, S., Scott, S. D., & Greenberg, S. (2004). Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. *Computer Supported Cooperative Work, 13*(5–6), 501–537.

Krumm, J., & Hinckley, K. (2004). The NearMe wireless proximity server. *Sixth International Conference on Ubiquitous Computing* (*Ubicomp 2004*), 283–300.

Krumm, J., & Horvitz, E. (2004). Locadio: Inferring motion and location from Wi-Fi Signal Strengths. *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (*Mobiquitous 2004*), 4–13.

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., et al. (2005). Place lab: Device positioning using radio beacons in the wild. *Pervasive Computing*, 3468/2005, 116–133.

LaViola, J., & Zeleznik, R. (2004). Mathpad2: A system for the creation and exploration of mathematical sketches. *ACM Transactions on Graphics* (*Proceedings of SIGGRAPH 2004*), 432–440.

Lee, S. W., & Mase, K. (2002). Activity and location recognition using wearable sensors. *IEEE Pervasive Computing, 1*(3), 24–32.

Lester, J., Choudhury, T., & Borriello, G. (2006). A practical approach to recognizing physical activities. *Proceedings Pervasive Computing 2006*, 1–16.

Letchner, J., Fox, D., & LaMarca, A. (2005). Large-scale localization from wireless signal strength. *AAAI*, 15–20.

Li, S. Z., & Jain, A. K. (2005). *Handbook of Face Recognition.* New York: Springer.

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

MacKenzie, I. S. (1992). Fitts' Law as research and design tool in human computer interaction. *Human-Computer Interaction, 7*, 91–139.

MacKenzie, I. S., & Ware, C. (1993). Lag as a determinant of human performance in interactive systems. *Proceedings of the SIGCHI conference on human factors in computing systems*, 488–493.

Maes, P., Darrell, T., Blumberg, B., & Pentland, A. (1995). The ALIVE system: Full-body interaction with autonomous agents. *Proceedings of the Computer Animation*, 11.

Matsushita, N., & Rekimoto, J. (1997). HoloWall: Designing a finger, hand, body and object sensitive wall. *Proceedings of the ACM Symposium on User Interface Software and Technology* (*UIST*), 209–210.

McNeill, D. (1992). *What gestures reveal about thought*. Chicago: The University of Chicago Press.

Mitchell, T. M. (1997). *Machine learning.* Boston: McGraw-Hill.

Mozer, M. C. (2005). Lessons from an adaptive house. In D. Cook & R. Das (Eds.), *Smart environments: Technologies, protocols and applications* (pp. 273–294). Hoboken, NJ: J. Wiley & Sons.

Nachman, L., Kling, R., Huang, J., & Hummel, V. (2005). The Intel mote platform: A Bluetooth-based sensor network for industrial monitoring. *Fourth International Symposium on Information Processing in Sensor Networks*, 437–442.

Neal, R., & Hinton, G. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Cambridge, MA: MIT Press.

Nielsen, J. (1994). *Usability engineering*. San Francisco: Morgan Kaufmann.

Norman, D. A. (2003). *Emotional design: Why we love (or hate) everyday things*. New York: Basic Books.

Oliver, N., & Horvitz, E. (2005). Selective perception policies for guiding sensing and computation in multimodal systems: A comparative analysis. *Computer Vision and Image Understanding, 100*(1–2), 198–224.

Oviatt, S. L. (2002). Breaking the robustness barrier: Recent progress on the design of robust multimodal systems. In M. Zelkowtiz, *Advances in Computers, Volume 56* (pp. 305–341). : Academic Press.

Patten, J., Ishii, H., & Pangaro, G. (2001). Sensetable: A wireless object tracking platform for tangible user interfaces. *Proceedings of the*

Conference on Human Factors in Computing Systems (*SIGCHI*), 253–260.

Pentland, A. (2004). Social dynamics: Signals and behavior. *Third International Conference on Development and Learning*, 263–267.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 84–91.

Picard, R. W. (2000). *Affective computing*. Cambridge, MA: The MIT Press.

Pickering, J. (1986). Touch-sensitive screens: The technologies and their application. *International Journal of Man-Machine Studies, 25*(3), 249–269.

Platt, J. (1999). Using analytic QP and sparseness to speed training of support vector machines. *Advances in Neural Information Processing Systems, 11*, 557–563.

Rabiner, L., & Juang, B. (1993). *Fundamentals of speech recognition.* Englewood Cliffs, NJ: Prentice Hall.

Rabiner, L. R. (1989). A tutorial in hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257–386.

Rekimoto, J. (2002). SmartSkin: An infrastructure for freehand manipulation on interactive surfaces. *Proceedings of the SIGCHI conference on human factors in computing systems*, 113–120.

Rekimoto, J., & Ayatsuka, Y. (2000). CyberCode: Designing augmented reality environments with visual tags. *Proceedings of the Designing Augmented Reality Environments* (*DARE 2000*), 1–10.

Reynolds, C. (2001). *The sensing and measurement of frustration with computers*. Unpublished master's thesis. Media Laboratory, Massachusetts Institute of Technology, Cambridge.

Rubine, D. (1991). Specifying gestures by example. *Computer Graphics, 25*(4), 329–337.

Rui, Y., & Florencio, D. (2003). New direct approaches to robust sound source localization. *Proceedings of IEEE International Conference on Multimedia Expo*, 737–740.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, & B. Yu, (Eds.), *Nonlinear estimation and classification*, New York: Springer.

Schmidt, A. (2000). Implicit human interaction through context. *Personal Technologies, 4*(2&3), 191–199.

Schmidt, A., Beigl, M., & Gellersen, H. W. (1999). There is more to context than location. *Computers & Graphics Journal, 23*(6), 893–902.

Sears, A., Plaisant, C., & Shneiderman, B. (Eds.). (1992). A new era for high precision touchscreens. In (Eds.), *Advances in human computer interaction.* Norwood, NJ: Ablex.

Shell, J., Vertegaal, R., Cheng, D., Skaburskis, A. W., Sohn, C., Stewart, A. J., et al. (2004). ECSGlasses and EyePliances: Using attention to open sociable windows of interaction. *Proceedings of the ACM Eye Tracking Research and Applications Symposium*, 93–100.

Shen, C., Lesh, N., & Vernier, F. (2003). Personal digital historian: Story sharing around the table. *ACM Interactions, 10*(2), 15–22.

Shen, C., Vernier, F. D., Forlines, C., & Ringel, M. (2004). DiamondSpin: An extensible toolkit for around-the-table interaction. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (*SIGCHI*), 167–174.

Smith, A., Balakrishnan, H., Goraczko, M., & Priyantha, N. (2004). Tracking moving devices with the Cricket location system. *Proceedings of the 2nd USENIX/ACM MOBISYS Conference*, 190–202.

Smith, J. R. (1999). *Electric field imaging*. Unpublished doctoral dissertation. Media Laboratory, Massachusetts Institute of Technology, Cambridge.

Stork, D. G. (1998). *HAL's legacy: 2001's computer as dream and reality.* Cambridge, MA: MIT Press.

Sugiura, A., & Koseki, Y. (1997). A user interface using fingerprint recognition—holding commands and data objects on fingers. *Proceedings of the Symposium on User Interface Software and Technology*, 71–79.

Tan, H. Z., Slivovsky, L. A., & Pentland, A. (2001). A sensing chair using pressure distribution sensors. *IEEE/ASME Transactions on Mechatronics, 6*(3), 261–268.

Tapia, E. M., Intille, S., & Larson, K. (2004). Activity recognition in the home setting using simple and ubiquitous sensors. *Proceedings of Pervasive 2004*, 158–175.

Tashev, I., & Malvar, H. S. (2005). A new beamformer design algorithm for microphone arrays. *Proceedings of International Conference of Acoustic, Speech and Signal Processing*, 101–104.

Tenneson, D., & Becker, S. (2005). ChemPad: Generating 3D molecules from 2D sketches. *SIGGRAPH 2005 Extended Abstracts*.

Tobii Technology. http://www.tobii.com. Retrieved February 10, 2007.

Tomasi, C., Rafii, A., & Torunoglu, I. (2003). Full-size Projection Keyboard for Handheld Devices. *Communications of the ACM, 46*(7), 70–75.

Ullmer, B., Ishii, H., & Glas, D. (1998). MediaBlocks: Physical containers, transports, and controls for online media. *Computer Graphics, 32*, 379–386.

Vogel, D., & Balakrishnan, R. (2004). Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. *Proceedings of the ACM Symposium on User Interface Software and Technology* (*UIST 2004*), 137–146.

Vogel, D., & Balakrishnan, R. (2005). Distant freehand pointing and clicking on very large high resolution displays. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 33–42.

Vranish, J. M., McConnell, R. L., & Mahalingam, S. (1991). "Capaciflector" collision avoidance sensors for robots. *Robotics research at the NASA/Goddard Space Flight Center, 17*(3), 173–179.

Want, R., Fishkin, K. P., Gujar, A., & Harrison, B. L. (1999). Bridging physical and virtual worlds with electronic tags. *SIGCHI '99*, 370–377.

Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems (TOIS), 10*(1), 91–102.

Ward, A., Jones, A., & Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications, 4*(5), 42–47.

Ware, C. (1990). Using hand position for virtual object placement. *Visual Computer, 6*(5), 245–253.

Ware, C., & Jessome, D. R. (1988). Using the Bat: A six-dimensional mouse for object placement. *Proceedings of the IEEE Computer Graphics and Applications*, 65–70.

Wayman, J., Jain, A., Maltoni, D., & Maio, D. (Eds.). (2004). Biometric systems: Technology, design and performance evaluation. New York: Springer.

Welch, G., & Bishop, G. (2004). *An introduction to the Kalman Filter*. (Dept. Computer Science Technical Report TR-95-041). Chapel Hill, NC: University of North Carolina at Chapel Hill.

Wellner, P. (1993). Interacting with paper on the DigitalDesk. *Communications of the ACM, 36*(7), 87–96.

Wilson, A. D. (2005). PlayAnywhere: A compact tabletop computer vision system. *Proceedings of the 18th Annual ACM Symposium on User Interface Software Technology*, 83–92.

Wilson, A. D., & Pham, H. (2003). Pointing in intelligent environments with the WorldCursor. *Proceedings of Interact*, 495–502.

Wilson, A. D., & Shafer, S. (2003). XWand: UI for intelligent spaces. *Proceedings of SIGCHI*, 545–552.

Wilson, D. H., & Atkeson, C. G. (2005). Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. *Proceedings of Pervasive 2005*, 62.

Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1995). Pfinder: Real-time tracking of the human body. *Proceedings of SPIE Photonics East*, 89–98.

Wu, M., & Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of the 16th Annual Symposium on User Interface Software and Technology*, 193–202.

Wu, Y., Toyama, K., & Huang, T. (2000). Wide-range, person- and illumination-insensitive head orientation estimation. *Proceedings of the International Conference on Face and Gesture Recognition*, 183.

Zhai, S., & Bellotti, V. (Eds.). (2005). Special Issue on Sensing-Based Interaction. *ACM Transactions on Computer-Human Interaction*.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the SIGCHI conference on human factors in computing systems*, 246–253.

Zhang, Z., & Hershey, J. (2005). *Fusing array microphone and stereo vision for improved computer interfaces* (Microsoft Research Technical Report MSR-TR-2005-174) Redmond, WA: Microsoft Corporation. Smart Technologies, Incorporated. http://www.smarttech.com. Retrieved February 10, 2007.

Zhao, W., Chellappa, R., Phillips, R. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys, 35*(4), 399–458.

11

VISUAL DISPLAYS

Christopher Schlick, Martina Ziefle, Milda Park, and Holger Luczak
RWTH *Aachen University*

Introduction **202**
Basic Principles and Concepts **202**
Image Generation 202
Electronic Information Displays 202
Display Segmentation 203
Display Dimensionality 203
Human-Related Display Functionality and Other Criteria 204
Quality Criteria **205**
Display Colors 205
Brightness and Glare 207
Resolution 208
Contrast and Gray Scales 208
Refresh Rates 209
Technologies **210**
Classification of Electronic Information Display Technologies 210
Direct-view Displays 210
Projection Displays 210
Off-screen Display Systems 210
Cathode Ray Tube 210
Dot-mask Screen 211
Slit mask (Trinitron Technology) 211
Slot-mask Screen 211
Linearity Error 212
Convergence Error 212
Moiré 212
Non-emitter/Liquid-Crystal Display 212
Plasma Displays 213
Electroluminescent Displays 214
Cathodoluminescent Displays 214
Laser Displays 215
Electronic Ink and Electronic Paper 216
Visual Performance **217**
Task Demands 218
Measures Used for Visual Evaluations 218
Effects of Specific Display Factors on Visual Performance 219
Comparison of Different Display Types 221
Standards, Rules and Legal Regulations and Health Protection Aspects **222**
Displays for Selected Applications **222**
Virtual Reality 222
Augmented Reality 223
Mobile Phones and Handheld Devices 224
References **225**

INTRODUCTION

Since the existence of mankind, humans have been inventing and building tools to make life easier and more comfortable. One of mankind's first tools with a visual display was the sundial, which was invented more than 3,000 years ago in babylon. Due to its basic physical principle, one of its significant disadvantages was that it could not show the time during the night or when the weather was cloudy. Moreover, the first sundials were in public places, so that people had to make the effort to go there in order to check the time. Later developments of clocks (also in public places) eliminated the disadvantages of the sundial. Church clocks could not only show the time under any weather condition and at any time of day or night, but they could also display the time acoustically and therefore bridge distances up to a few miles. The following developments are well known–from the grandfather clock and the fob watch continuing with the first analog and digital wristwatches to today's high performance wrist computers with high-resolution wrist-worn visual displays (Luczak & Oehme, 2002). They do not only display the time accurately, but they are truly multifunctional. For instance, some can visualize individual heart-rate profiles, and others can be used to chart one's precise location on earth with the help of the global-positioning system. A few can even replace personal digital assistants.

BASIC PRINCIPLES AND CONCEPTS

Image Generation

Different visual displays rely on different physical principles to generate an image: light can be emitted, transmitted, or reflected by the display. Examples from noncomputer displays might be helpful to explain the basic principles and point out relevant advantages and restrictions.

Writing on a piece of paper alters the reflective properties of the paper from a highly reflective white to a less reflective blue or black. Ambient light is needed to read what is written on the paper, but the contrast ratio between text and background remains the same under different lighting conditions. The same principle holds for reflectance-based displays such as liquid crystal displays (LCDs) without backlighting or electronic ink and paper (see sections "Nonemitter/Liquid-Crystal Display" and "Electronic Ink and Electronic Paper.")

Transparency film used with an overhead projector and slides used with a projector are examples of transmission. Different parts of the transparency or slide transmit light of different wavelengths (i.e. color) in different amounts. Although transmission is used as a basic principle in many electronic information displays, it is also often combined with other physical principles. For instance, a front-projection display (see section "Classification of Electronic Information Display Technologies") consists of a light source, an LCD generates the image and transmits the rays of light, and finally the projection surface reflects the light into the eye of the observer.

A lighthouse is an example of emission. Whereas its light can be easily seen at night, it is barely visible in bright daylight. Examples of emission-based displays are CRTs (see section "Cathode Ray Tube") electroluminescent displays (see section "Electroluminscent Displays"), and cathodoluminescent displays (see section "Cathodoluminescent Displays"). Similar to a lighthouse, these displays need to be brighter than the ambient light to be perceived properly.

Electronic Information Displays

The development of advanced display technologies began with the cathode ray tube or CRT, which was first discovered in the nineteenth century, although the observation of a glow from the electronic excitation of gas in an evacuated tube may go back as early as the 17th century. The invention of the device itself is generally attributed to Karl Ferdinand Braun. The "Braun tube" reportedly first built in Strasbourg, Germany, in 1896 or 1897, used both deflection and a fluorescent material for the screen. It was probably the first application of an electronic information display in natural sciences (Castellano, 1992). CRT displays have dominated the market for many years since their invention. However, the increasing requirements of consumers have led to rapid developments of alternative concepts for electronic information displays. Today, a large variety of display technologies are competing with CRTs in terms of image quality, and new technologies, which are currently under development, may soon cause more major changes in display demand (see section "Classification of Electronic Information Display Technologies").

There are two basic methods for displaying information visually: (a) digital and (b) analog. A digital display uses binary numbers of arbitrary length for displaying symbols such as characters or icons; whereas an analog system uses a continuous signal spectrum for information presentation. If an instant impression is required, analog displays often present information better. Many people glance quickly at their analog watch and know roughly what the time is or at an automobile dashboard and know that they are driving too fast. Analog displays translate a value of a continuous variable into an angle or a length. Analog displays used for control devices consist of a scale and an indicator or hand. Either the scale or the hand moves. There are a number of guidelines for the design of analog displays (Woodson, 1987; Baumann & Lanz 1998).

When accuracy is a critical issue, however, digital displays are preferred. Reading analog meters accurately requires time and cognitive skills, whereas writing down the value on a digital display is merely a case of copying down the numbers. In cases where both accuracy and quick reckoning are required, hybrid displays are often used.

Because a computer is a digital device, all data displayed are binary numbers, and therefore all commonly used electronic information displays are digital displays. Nonetheless, some application programs mimic analog display devices because of their ergonomic advantages. If the spatial resolution and the refresh rate of the digital display are sufficiently high and the covered color space is sufficiently large, no significant differences in visual performance from an analog display occur (see section "Resolution").

Display Segmentation

Common personal computer (PC) displays have a sufficiently high resolution to display a virtually unlimited set of characters, symbols, and graphics. Conversely, a wide variety of displays with lower spatial resolution is found on other kinds of computerized technical devices, such as mobile phones or music players. Here, the set of displayable tokens is often far more restricted.

Basic display elements are binary, whether they are on or off. A ternary display element can be built from a two-color light-emitting diode displays (LED). To transmit more information, display elements with more discrete states are necessary using i.e., different colors or brightness levels to distinguish between states.

A more common way to increase the amount of displayable information is the grouping of a number of binary display elements into a unit. The classical seven-segment display (Fig. 11.1) and many custom displays are examples of this approach.

If multiple binary display elements are arranged in matrix form and the elements are enhanced to display multiple colors and different levels of brightness, the basic concept of the common CRT and flat-panel displays can be derived. The shape of picture elements (pixels) in current electronic information displays is typically square or rectangular. The pixels are arranged in horizontal rows and vertical columns, which are called "pixel matrices" or "pixel formats."

Display Dimensionality

Visual displays can be distinguished by means of the spatial dimensionality of the image generated. Clearly, as the price of the display and the complexity of the display technology increase, the more dimensions are used.

The simplest way to display information is to rely on only one spatial dimension. Although the notion of one dimensional (1D) display is not strictly correct since human perception requires a display unit to have two spatial dimensions, the secondary dimension of such displays does not provide information and is simply a function of the primary dimension. 1D display represents the minimal approach to presenting information, which is encoded by either discrete or continuous variables in one dimensional states space. Nevertheless, a simple source of light, such as a bulb or a candle, can convey a binary message of arbitrary length. For most current applications, for example, to show the status of a device as "on" or "off," LEDs are used. LEDs give off light radiation when biased in the forward direction. Most light emitting diodes function in the near infrared and visible ranges, with the exception of ultra violet LEDs. When displaying a continuous variable in one dimension, a control window usually indicates the actual level of the parameter, which is measured or controlled. The volume level of a speaker or the temperature in a room can be indicated using a row of pixels, LEDs, or a slider on a scale.



FIGURE 11.1. Seven-segment display.

In order to display more complex information, which can only be shown using two independent spatial dimensions (length and width), two-dimensional (2D) displays are used. This category of displays has by far the biggest segment in the market. They are suitable for displaying all kinds of graphical information. Images are formed by activating pixels, each of which has a unique location (x, y) within a two-dimensional plane, and a color and gray scale value may be assigned to each. Although these displays have only two dimensions, stereoscopic perception is also feasible using depth cues such as relative size, height relative to the horizon, interposition or occlusion, shadows and shading, spatial perspective, linear perspective, and texture gradients. Even more realistic three-dimensional (3D) scenes can be viewed on 2D screens by applying special viewing devices, such as shutter glasses or polarizing glasses. In order to simulate depth perception, each of the eyes must see a slightly different image (binocular parallax). For instance, the so-called shutter glasses alternately block each eye's view so that each perceives only the image intended for it. The alternation of images occurs many times per second and these separate images are fused into one true 3D image at higher cortical levels. The speed of the shutters is directly proportional to the refresh rate of the monitor. In the case of polarising glasses, different light polarisation modes are used to encode the information for the left and the right eye, respectively. The polarizing glasses filter out the unwanted photon polarization from each view and two separate images are projected onto the visual cortex. These techniques are used with immersive desks or a Computer Animated Virtual Environment (CAVE).

One of the limitations of such stereoscopic displays is that motion parallax, as one of the depth cues, is not provided automatically. Display devices that are able to provide the correct stereoscopic perspective to each of the viewer's eyes over a range of viewing positions can be defined as auto-stereoscopic. Volumetric 3D displays fall into this category. There is a growing need to present complex and voluminous 3D information in such way that it may be interpreted rapidly, naturally, and accurately. Volumetric displays (also called direct volume display devices) permit 3D-image data to be presented in a transparent volume. Innate depth cues inherent in three-dimensional objects are then automatically present, and, in principle, the three-dimensional images may be viewed from an arbitrary direction. Volumetric display systems provide a transparent volume in space where animated images may be depicted. Since these images have three physical dimensions, depth cues are inherent without the need to compute complex transformations, which is a considerable benefit of this type of display. Furthermore, images may be observed naturally without the need to wear special glasses. Common to all volumetric display systems is a volume or region occupying three physical dimensions within which image components may be positioned, depicted, and possibly manipulated. Volumetric display systems employ a method for image construction similar to 2D displays. Elements, which are called "voxels," are activated at the appropriate locations within the 3D space. Each voxel has a unique position (x, y, z) and can be assigned a color and gray-scale value. Depending on the

technology used to activate the voxels, the volumetric displays can be assigned to several categories, such as emissive volume display, varifocal mirror display, laser displays, or holographic displays. A more detailed description of volumetric display technology can be found in Blundel and Schwarz (2000).

Human-Related Display Functionality and Other Criteria

The paradigm of wearable computing requires the visual displays to be worn on the human body (Azuma, 1997; Feiner, 2002). A popular approach for mobile applications is to mount the display on the user's head. Thus, the user can work hands-free and is always able to perceive information in his or her field of vision. These displays are commonly referred to as head-mounted displays (HMDs). From the technological point of view, HMDs can roughly be split into three main categories, according to the way the image is provided to the user (Fig. 11.2).

The first category, referred to as "screen-based HMDs," comprises all HMDs whose picture elements are created in a spatially adjacent way. The ocular image forming displays use technologies such as CRT, LCD, digital mirror devices (DMD), or organic light-emitting diodes (OLEDs). Most of today's HMDs on the market, however, are based on transmissive or reflective liquid-crystal minidisplays (LCD) (Holzel, 1999). Miniature LCDs are available for relatively low prices and provide an appropriate resolution (SXGA resolution of up to 1280 × 1024 pixels), while being lightweight (von Waldkirch, 2005).

In the retinal-projection method, the display image is projected directly onto the retina in the same way as a slide is projected onto the screen. Projection displays are normally designed in the form of a Maxwellian-view optical system (Bass, 1995), where the screen plane is optically conjugated to the retina and the illumination source is conjugated to the eye's pupil plane. Consequently, such displays can only be implemented on the basis of illuminated screens (like LCD and DMD), and not with self-emitting technologies like OLED and CRT (von Waldkirch, 2005).

Scanning displays, where the image is scanned pixel by pixel directly onto the retina, are an alternative to screen-based displays. Retinal scanning displays (RSDs), also referred to as virtual retinal displays (VRDs), are most important in this category. VRD technology was first proposed in 1992 by Sony. Since 1992, researchers at the Human Interface Technology Lab (Washington, DC) have been developing this technology to obtain a commercial product. In 2003, they presented the first commercial VRD, called "Nomad," together with the U.S.-based company Microvision. With this technology, an image is scanned directly onto a viewer's retina using low-power red, green, and blue light sources, such as lasers or LEDs (Urey, Wine, & Lewis, 2002). The VRD system has superior brightness and contrast compared to LCDs and CRTs, as it typically uses spectrally pure lasers as its light source (Stanney & Zyda, 2002). Finally, a combination of a scanning technology and a screen-based system is possible (see Pausch, Dwivedi, & Long, 1991; Aye, Yu, Kostrzewski, Savant, & Jannson, 1998; Fruehauf, Aye, Yua, Zou, & Savant, 2000).

Beside technological categorization, the various HMD concepts can be classified according to their functionality or intended purpose. Fig. 11.3 shows a rough functional categorization of HMDs (von Waldkirch, 2005). Here, HMDs can be divided into monocular, binocular, and binocular displays. In addition, all these types can provide the image in either a closed-view or see-through mode.

Monocular displays have one display source and thus provide the image to one eye only. Therefore, in comparison with biocular or binocular HMDs, monocular displays are lighter and

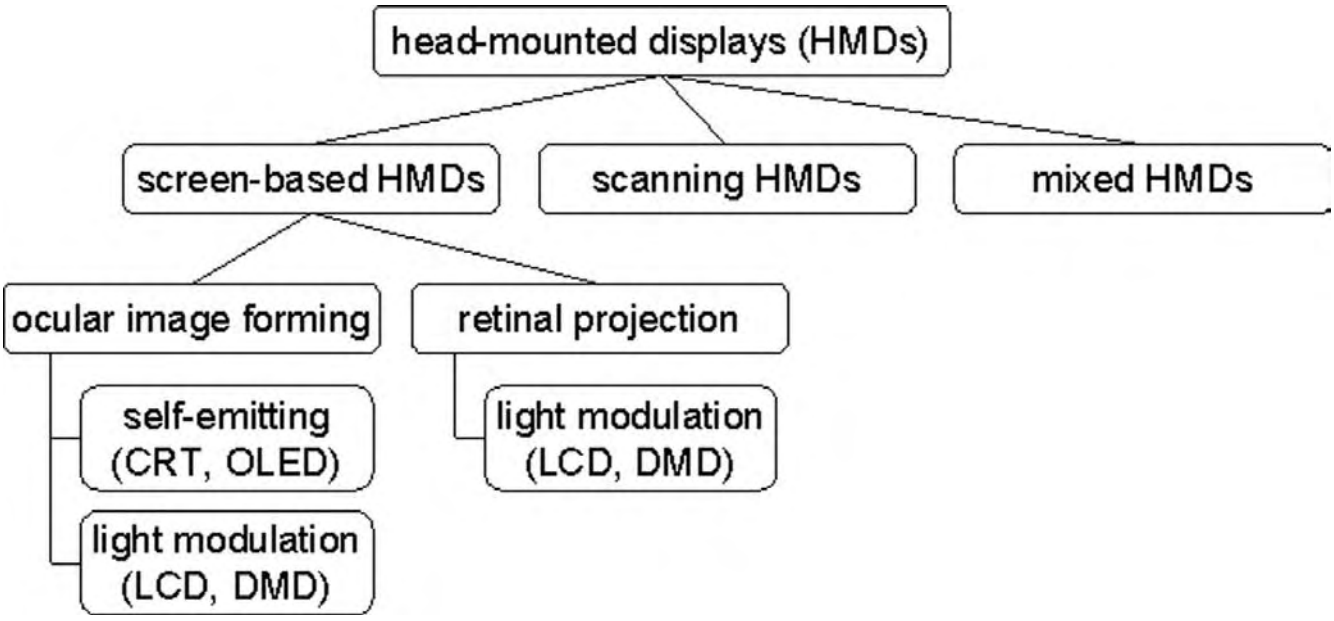


FIGURE 11.2. Systematics of head-mounted displays (von Waldkirch, 2005).

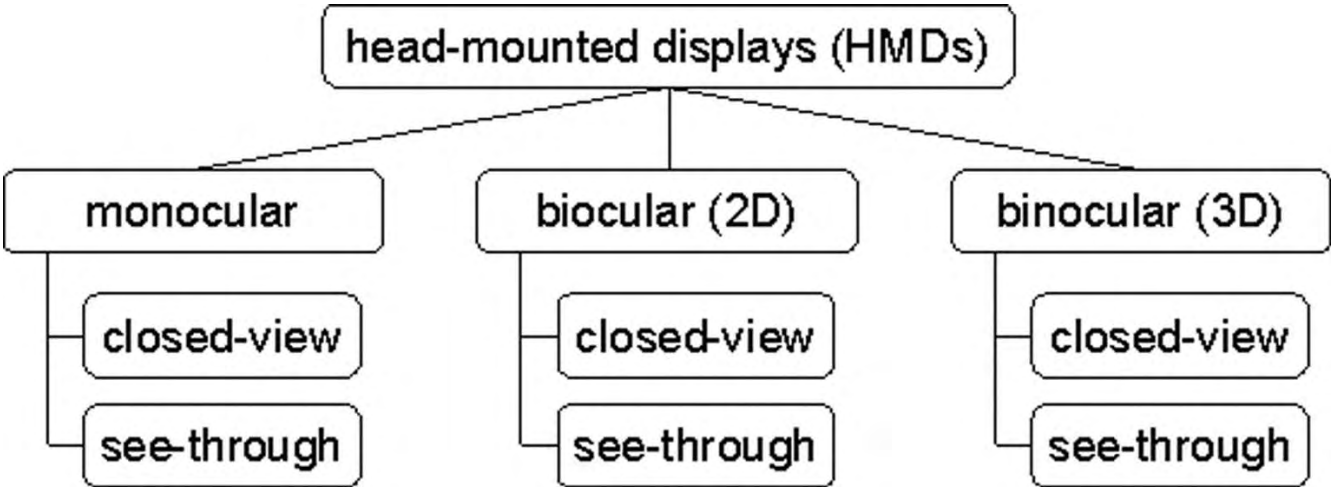


FIGURE 11.3. Head-mounted displays categorized according to functions (von Waldkirch, 2005).

cheaper. However, in a monocular HMD, the view seen by the eye, which is not blocked by a screen, may produce binocular rivalry with the image seen through the HMD. Biocular HMDs have two displays with separate screens and optical paths, allowing both eyes to see exactly the same images simultaneously. Thus, the user perceives a 2D image only, similar to a computer screen. Binocular displays allow stereoscopic viewing with 3D-depth perception. To produce the stereoscopic view, two spatially slightly incongruent images are provided to the left and to the right eye (binocular parallax, see section "Human Related Display Functionality and Other Criteria").

Beside the HMDs, important categories of displays are wrist-worn and handheld displays. These displays are very widespread, because they are integrated in wristwatches, mobile phones, personal digital assistants (PDAs), media players, and other portable devices (see section "Mobile Phones and Handheld Devices").

A truly "wearable" visual display was introduced recently by Philips Research and Textile Institute TITV Greiz. The company has been developing photonic textiles–fabrics that contain lighting systems and therefore can serve as electronic information displays. Multicolored light-emitting diodes (LEDs) were successfully integrated into fabrics without compromising the softness of the cloth. Passive matrixes of 10 10 red, green, and blue LED packages were placed on interconnecting substrates, made entirely of cloth. Applied in soft fabric, the light from the smallest pixels diffuses, resulting in a more or less continuous light-emitting surface. Photonic textiles can also be made interactive. Philips has achieved interactivity by incorporating sensors (such as orientation and pressure sensors) and communication devices (such as Bluetooth, GSM) into the fabric. Photonic textiles open up a wide range of possible applications in the fields of communication and personal health care (Philips Research Press Information, 2005).

QUALITY CRITERIA

Display Colors

Different displays can present differing numbers of colors. The simplest displays have two pixel states, either on or off and, therefore, are monochrome. The greater the difference between these two states, expressed as a contrast ratio or the weighted wavelength differences of the light falling upon the observer's eye, the more easily the viewer can distinguish between them. More complex displays, called gray-scale displays, are also able to present different luminance levels of pixels.

Clearly, all modern electronic information displays reproduce color as well (Hanson, 2003). The Commission International de l'Eclairage (CIE) chromaticity diagram is often used to represent a display's color capabilities. The diagram is based on the CIE standard *XYZ* system. *X*, *Y*, and *Z* are hypothetical primary colors. Fig. 11.4 depicts the equal energy-matching functions of the standard *XYZ* system. The *XYZ* primaries were chosen so that whenever equal amounts of them are combined, they match white light.

Independently of the absolute value of *X*, *Y*, and *Z*, relative amounts—denoted by lowercase letters—can be used to describe a color. Because x, y, and z are relative amounts, $x + y + z = 1$. Consequently, when x and y are known, z is known as well, because $z = 1 - (x + y)$. Therefore, a curve can be used to depict all the possible combinations of the *XYZ* primaries, as shown here in Fig. 11.5. At the intersection of $x = 0.33$ and $y = 0.33$ pure white (i.e. achromatic light) can be found. The curved line, the locus of spectral colors, represents the x, y, and z values for all of the spectral colors. The hue of any color within the curved line can be determined by drawing a line from pure white

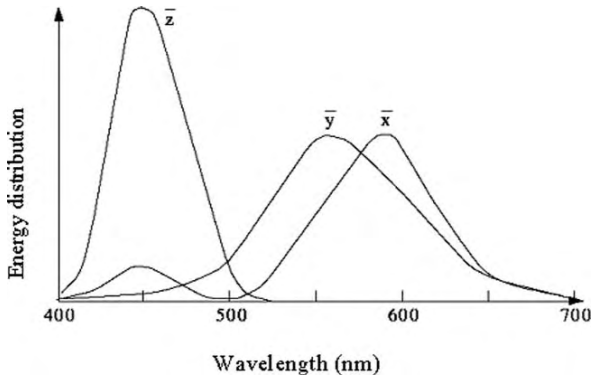


FIGURE 11.4. Equal energy-matching functions of the standard XYZ system. The curves $\bar{X}$, $\bar{Y}$, and $\bar{Z}$ and show the relative amounts of the X, Y and Z primaries needed to match the color of the wavelength of light (cf. Kaufman, 1974).

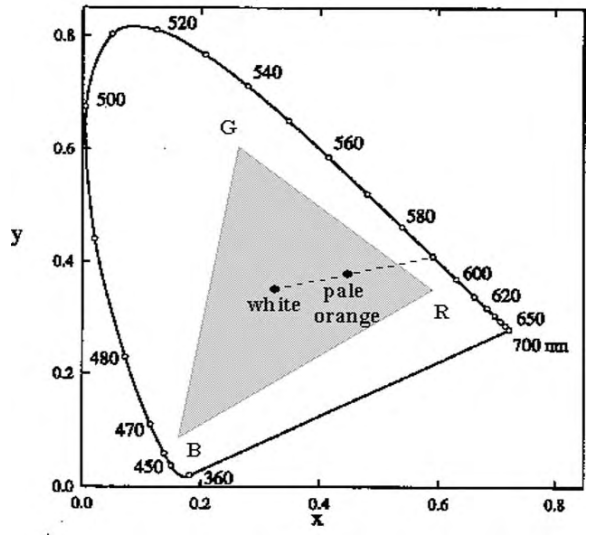


FIGURE 11.5. The CIE chromaticity diagram. The curve, the locus of spectral colors, represents the *x*, *y* and *z* values for all of the spectral colors between 380 nm and 770 nm. The triangle depicts the colors that can be presented by a display that uses the three primary colors at the corners of the triangle. The hue of any color within the curved line can be determined by drawing a line from pure white ($x = 0.33$ and $y = 0.33$) through the color of interest to the locus line. A pale orange ($x = 0.45$ and $y = 0.35$) is marked by a black dot as an example. The saturation is given by the relative length of the dashed line between white and the color of interest and white and the intersection of this line with the locus (cf. Kaufman, 1974).

through the color of interest to the locus line. The saturation is given by the relative length of the line between white and the color of interest, and white and the intersection of this line with the locus (see Fig. 11.5).

The representation of this information in the computer's hardware and software places a further restriction on the variety of colors. Popular configurations devote 4 to 32 bits to the representation of color information, yielding 16 to 16,777,216 colors with different hues and saturation.

Because *x* and *y* are relative values, an additional parameter, not shown in the Fig. 11.5, is needed to specify the luminance. It is denoted by *Y*, and hence this system of color characterization is referred to as the "*xyY*-system."

In order to depict a great variety of different colors, common displays use the three primary colors: red, green, and blue. The actual scope of colors is restricted by the position of each of the three primaries in the CIE chromaticity diagram and lies within the triangle defined by these primaries (see Fig. 11.5).

Whereas the CIE chromaticity diagram is useful in depicting the color space, additional systems are used to describe color in the context of human-computer interaction (HCI). The specification of *hue, saturation, and brightness* (HSB; sometimes instead of brightness the terms *luminance* or *lightness* are used) provides a device-independent way to describe color. For this model, the color space is an upside-down cone (see Fig. 11.6). On the edge of the cone base, the visible light spectrum is arranged in a circle by joining red and violet. Hue is the actual color and can be specified in angular degrees around the cone starting and ending at red = 0° or 360°, in percent or eight-bit values (0 to 255). Saturation is the purity of the color, measured in percent or eight-bit values from the center of the cone (min.) to the surface (max.). At 0% saturation, hue is meaningless. Brightness is measured in percent or eight-bit values from black (min.) to white (max.). At 0% brightness, both hue and saturation are meaningless.

RGB stands for the three basic colors—red, green and blue—that are produced by the visual display. A number of other colors can be produced by *additive color mixing*. If any two of the color channels are mixed in equal proportions, new colors are

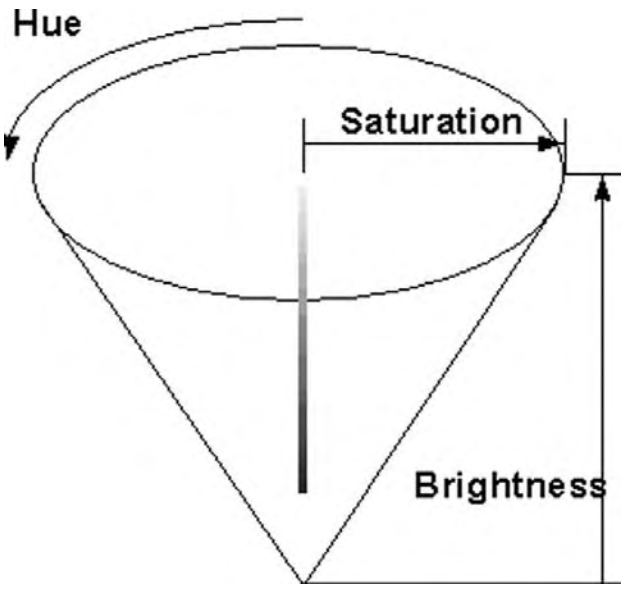


FIGURE 11.6. Hue, saturation and brightness system for specifying color.

created: blue and green combine to create cyan (bright, light blue); red and blue make magenta (a bright pink); and red and green make yellow. If all three colors are mixed together equally at full power, the result is white light. Colors can be specified by giving the relative power of each color in percent or as an 8-bit number (0 to 255 in decimal or 00 to FF in hexadecimal). So, for example, (0, 0, 0) specifies black; (255, 0, 0), a pure red and (255, 255, 255), white.

CMYK model defines cyan, magenta, yellow, and black as main colors. This system is used to specify colors on the monitor for printing. In printing, colors are mixed subtractively and, using red, green, and blue, it would not be possible to produce many colors. By choosing cyan, magenta, and yellow as basic colors, many other colors, including red, green, and blue, can be produced. Theoretically, when all three basic colors are printed over each other, the resulting color should be black. In practice, this is not the case, and a fourth printing process with black ink is also used. Values for CMYK are often specified as percentages.

Brightness and Glare

Brightness can be defined as the perceived amount of light that comes from an object. It is a physiological interpretation of luminance (L), an important photometric quantity. It is a measure of the luminous intensity of light (I) emitted from a light source per unit surface area (A) normal to the direction of the light flux (Çakir, Hart, & Stewart, 1979). In this context, θ is the angle between the perpendicular from the surface and the direction of measurement (Boff & Lincoln, 1988):

$$L = \frac{I}{A}\cos\theta \tag{11.1}$$

The unit of luminance is lumen per steradian per square meter or candela per square meter (cd/m^2). In some cases the unit foot-lambert (fL) is used to describe luminance (1 fL = 3,426 cd/m^2).

The variable I is the luminous intensity (also called "radiant intensity"), which represents the emitted flux-per-unit solid angle Ω:

$$I 5 \frac{d\Phi}{d\Omega} \tag{11.2}$$

Luminous intensity is measured in candela (cd); the unit is lumen per steradian.

Brightness of a display depends not only on the optical power generated, transmitted, or reflected by the display but also on the response of the human eye at certain wavelengths. The measure for this relation is photopic luminosity, $K(\lambda)$, which relates optical power in watts at a given wavelength to its effect on the human visual system. This luminosity can be considered as the optical spectral response of the eye of a "standard observer" (Nelson & Wullert, 1997). Mathematical relation of luminance to the spectral distribution of optical power $P(\lambda)$ can be expressed using the following equation, in which the integration is performed over the range of visible wavelengths (Macadam, 1982):

$$\Phi = \int_{\lambda=380}^{\lambda=770} K(\mathrm{l})P(\mathrm{l})d\mathrm{l} \tag{11.3}$$

The human eye cannot collect all of the light that is radiated or reflected from the source. Brightness also depends on the size of the surface spot, which the light is emanating from. Figure 11.7 shows how the brightness of a surface is actually given by the luminous flux per unit of the projected area of the emitting surface per unit solid angle depending on the viewing angle.

Some displays, such as LCDs, appear dimmer from an oblique angle than from the normal viewing angle; whereas most emissive displays, such as CRTs, emit light in such a way that the angular luminous intensity approximately follows Lambert's cosine law (Lambertian surface), resulting in approximately constant luminance across all viewing directions.

High luminance levels in the field of view cause glare discomfort. Glare caused by light sources in the field of view is called "direct glare"; glare caused by light being reflected by a surface in the field of view is called "reflected glare" (Fig. 11.8).

Reflected glare can occur from specular (smooth, polished, or mirror-like) surfaces, spread (brushed, etched, or pebbled) surfaces, diffuse (flat or matt) surfaces or as a combination of the above three (compound) (Sanders & McCormick, 1993). Glare sources are more disturbing when they have higher luminance and when they are closer to the fixation point (Sheedy, 2005). Experiments also show that visibility is decreased by glare, and the decrease is greatest when the source of the glare is in the line of vision (Boff & Lincoln, 1988).

In order to avoid glare, it is advisable to position the display right-angled to the window (so that the line of vision is parallel to the window). The display can be protected with curtains, blinds, or movable walls. Lamps that can be reflected in the monitor must not have a mean luminance of more than 200 cd/m^2 and the maximum luminance must be less than 400 cd/m^2 according to German standard DIN 5035-7 (Fig. 11.9).

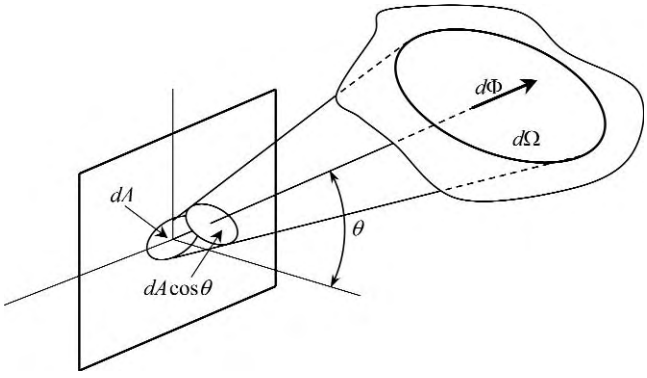


FIGURE 11.7. Definition of the brightness of a surface as a function of the direction from which the surface is observed (Nelson & Wullert, 1997).

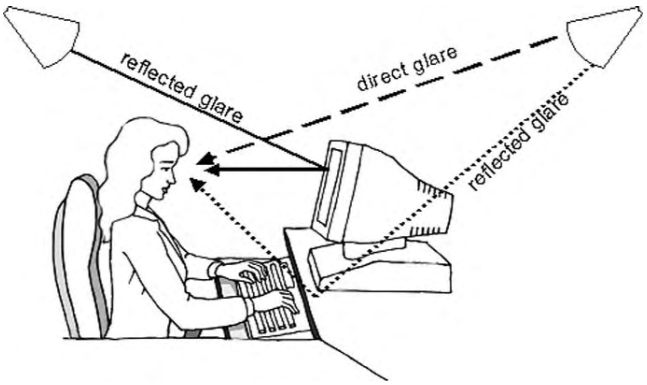


FIGURE 11.8. Direct and reflected glare.

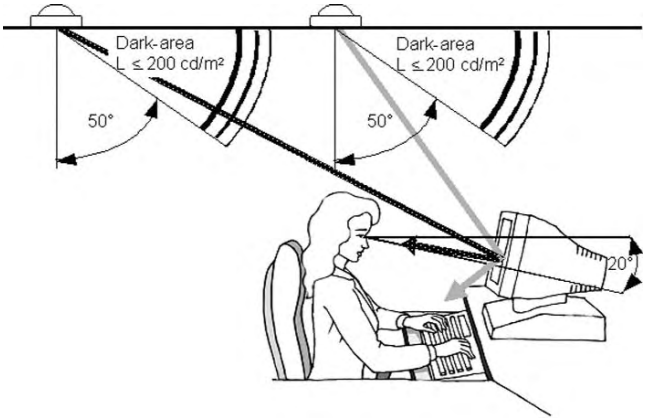


FIGURE 11.9. Glare can be avoided by limiting lamp luminance and by positioning the display correctly.

Resolution

The arrangement of pixels into horizontal rows and vertical columns is defined as the *pixel format.* It is often referred to as a "resolution." However, this is not a resolution parameter by itself. The resolution of a monitor mainly depends on its screen diagonal and its dot pitch (stripe pitch, SP). When set to lower resolutions, a pixel encompasses multiple dots. Thus, defining the resolution pixel density, or the number of pixels per linear distance (pixels per inch or pixels per centimeter) plays an important role. This parameter indicates how close the pixels are.

As a picture element, the pixel is the basic unit of programmable color on a display or in a digital image. In color CRT, the pixel is formed from a number of phosphor dots and may consist of a number of triads that are composed of a red, green and blue phosphor dots. The dot pitch in a CRT display with a shadow mask is defined as the distance between the holes in the shadow mask, measured in millimeters (mm). In desktop monitors, common dot pitches are 0.31 mm, 0.28 mm, 0.27 mm, 0.26 mm, and 0.25 mm. In matrix-driven, flat, panel displays, every single pixel is composed of a red, a green, and a blue phosphor dot or filter element, and dot pitch is defined as the center-to-center distance between adjacent green phosphor or filter element dots. Thus, the pixel density is the reciprocal of pixel pitch, which is equal to the dot pitch.

There is a simple relationship between resolution, in pixel/inch, and viewable diagonal screen size in inches for different monitor pixel formats. The resolution quality levels are defined as follows (Castellano, 1992):

- Low resolution: <50 pixels/inch
- Medium resolution: 51 to 70 pixels/inch
- High resolution: 71 to 120 pixels/inch
- Ultrahigh resolution: >120 pixels/inch

The highest resolution (x_{max} y_{max}) that can be represented on a CRT monitor can be calculated as follows:

$$x_{max} = \frac{D_{eff} \cos \alpha}{SP} \tag{11.4}$$

D_{eff} is the effective screen diagonal, where the picture can be viewed. The angle α (angle between the horizontal and the screen diagonal) can be calculated using knowledge about the aspect ratio between the vertical (V) and horizontal (H):

$$\alpha = \arctan\left(\frac{V}{H}\right) \tag{11.5}$$

As an example, a 17″ monitor (D_{eff} = 15,8 inches) with a horizontal stripe pitch of SP = 0.26 mm and an aspect ratio $H:V$ = 3:4 (common computer monitor) a maximum number of dots presentable in the horizontal of 1,235 or 926 dots in the vertical is possible. For our example, this means that with a resolution of 1,024 × 768 on the graphic card, each dot that has to be presented needs to be provided with at least one phosphorus wire. However, with a resolution of 1,280 × 1,024, not all dots can be represented clearly.

Contrast and Gray Scales

Strictly speaking, contrast is not a physiologic unit, but is nevertheless one of the most important photometry quantities. It is related to a lot of visual performance criteria, such as visual acuity, contrast sensitivity, speed of recognition, and so forth. There are different definitions of contrast.

The *Weberscher Contrast* C_w (luminance contrast) in accordance with the International Lighting Commission is defined as follows:

$$C_w = \frac{L_{object} - L_{background}}{L_{background}} \tag{11.6}$$

The definition implies that a negative luminance contrast is created by presenting a dark object on a bright background and a positive contrast is created by presenting a bright object on a dark background (Ziefle, 2002). This formula is not symmetric; if the luminance of the object is much greater than the background luminance, the numerical values of contrast are large

and increase rapidly. If the background luminance is much greater than the object luminance, the luminance contrast tends to move asymptotically to the value $C_w = -1$.

The *Michelson Contrast* C_m (also called "modulation contrast," "depth of modulation," or "relative contrast about the mean luminance") is generally used for periodic stimuli that deviate symmetrically above and below a mean luminance value, for example, gratings or bar patterns, and is computed as follows (Boff & Lincoln, 1988):

$$C_m = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}} \tag{11.7}$$

$L_{\max}$ and $L_{\min}$ are maximum and minimum luminance in the pattern. The Michelson Contrast will take on a value between 0 and 1. The minimum modulation threshold, which means the smallest brightness modulation that can be detected, occurs at a brightness modulation of approximately 0.003 at about 3 cycles per degree of the visual field (Nelson & Wullert, 1997).

Sanders and McCormick (1993) specified another popular possibility for defining contrast (called *luminous contrast*) as the difference between maximum and minimum luminance in relationship to maximum luminance:

$$C_l = \frac{L_{\max} - L_{\min}}{L_{\max}} \tag{11.8}$$

For a simple luminance increment or decrement relative to background luminance (such as a single point), the *contrast ratio* is used (Boff & Lincoln, 1988). This simple ratio of two levels of luminances is widely used in engineering (Bosman, 1989):

$$C_r = \frac{L_{\text{object}}}{L_{\text{background}}} \tag{11.9}$$

These contrast measures can be converted from one to another. For example, given the contrast ratio C_r, and knowledge of positive contrast conditions ($L_{\max} = L_{\text{object}}$; $L_{\min} = L_{\text{background}}$), the other visual contrasts can be calculated as follows:

Weberscher Contrast:

$$C_w = C_r - 1 \tag{11.10}$$

Michelson Contrast:

$$C_m = \frac{C_r - 1}{C_r + 1} \tag{11.11}$$

Luminous Contrast:

$$C_l = \frac{C_r - 1}{C_r} \tag{11.12}$$

In a computer display, the contrast ratio of symbols to background has to be more than 3:1. The signs have to be represented sharply up to the edges of the screen. Modern LCDs have a contrast ratio of 500:1 or more. Since LCDs do not emit light, the luminance in the previous equation for contrast refers to the luminance of light either passing through the display (for a backlit transmissive type) or the luminance of the light reflected off the display's surface (for a reflective type LCD). In multiplexed LCDs, the contrast is affected by the viewing angle. Therefore, the contrast should be indicated by referring to the solid angle, known as the "viewing cone" (Castellano, 1992).

Large contrast ratios are also necessary in order to satisfy gray-scale requirements. Based on the idea of the brightest areas being white and the darkest areas being black, levels of brightness in between the two extremes are referred to as gray levels or gray shades, and the ability to display them as gray scale (Nelson & Wullert, 1997). Technically, gray scale is a term that should be applied only to monochrome or "gray" displays. The term is now often applied to color displays where intermediate brightness controls are required by the system (Castellano, 1992). The number of gray scales is determined both by the contrast level and the ability of the human-visual system to distinguish between the different brightness levels. Our visual system reacts to the changes in brightness level as a logarithmic function; therefore very small differences in brightness might be not perceived (Theis, 1999). The acceptable difference in brightness levels between scales is 1.414 (the square root of 2). In order to obtain five levels of gray scale above background, a display must have a contrast ratio of at least 5.6:1 (1, 1.41, 2, 2.82, 4 and 5.65:1) (Castellano). Full-color displays have about 128 to 256 linear gray levels. The number of gray shades (G) that can be displayed can be defined as a logarithmic function based on contrast ratio (Nelson & Wullert):

$$G = 1 + \frac{1}{\log\sqrt{2}} \log\left(\frac{L_{\max}}{L_{\min}}\right) \tag{11.13}$$

Refresh Rates

The number of times that the image on the display is drawn per second is called the *display refresh rate* (also called *frame rate* or *vertical scanning rate*). The unit of measurement is Hertz (Hz). A high-display refresh rate prevents a flickering image, because there is only a small amount of time between two successive stimulations of a single dot. A refresh rate of between 70 and 80 Hz is needed to ensure a flicker-free image on the display (the flicker is not perceivable) (Bosman, 1989).

In CRT technology, the phosphorus dots (phosphorus stripes) are stimulated by the electron beam. Because they light up for only a fraction of a second, stimulation must occur several times a second. It passes line by line while the electron beam is writing from left to right and then returns to the beginning of the following line. Line build begins in the upper left corner, line by line, until each dot has been stimulated once. Then the electron beam goes back to the upper-left corner and begins to build up the picture again.

The line frequency (or horizontal frequency 2 measured in kHz) is another characteristic of CRT displays. It measures the number of lines the electron beam can draw per second. In this case, line frequency, display-refresh rate, and resolution are directly connected with each other. A monitor with a display-refresh rate of 70 Hz and a resolution of 1,024 × 768 needs to have an electron beam that is capable of drawing 70 × 768 lines = 53,760 lines per second. This means that the monitor has to process a line frequency of at least 53.76 kHz. If one wants to use a higher resolution, it is necessary to check whether the

monitor is capable of processing such a line frequency; otherwise damage might occur.

In addition to the frequencies described, the image representation mode can also lead to poor-quality representation. There are two image representation modes: interlaced and noninterlaced mode. With the interlaced mode, the image formation is divided into two half images. Here, the electron beam first builds all uneven lines and then all even lines.

This is the common process for televisions (for Phase Alternation Line, PAL); 50 half-images per second are normal, which means a vertical scanning frequency of 50 Hz but a full screen frequency of only 25 Hz. Nowadays, monitors with a non-interlaced mode are commonly used. In this case, the electron beam builds all of the lines one after another, without exception.

In general, active or light emitting displays, such as CRTs, can be written more quickly than passive or reflecting displays, such as LCDs. The higher speed is usually related to increased power consumption. Multiplexed LCDs have refresh rates of about 40 to 50 frames per second. The physics of the display is such that it takes some time to turn on any pixel or line. Recent advantages in LCD technology have reduced response time to about 8–20 ms.

TECHNOLOGIES

Classification of Electronic Information Display Technologies

Direct-view displays. These are video displays in which the light produced by a display device is viewed directly without first bouncing off a screen. All CRT, LCD, .plasma televisions, and computer monitors are direct-view displays. These displays tend to work best in bright-light conditions and have greater light output than projection displays.

Projection displays. Unlike a direct-view system, a projection display relies on the projection of an image onto a screen. There are front- and rear-projection systems, which mainly differ with regard to screen technology. The front projection utilizes a reflective screen surface while a rear projection uses a transmissive screen surface. Projection displays work best in dimly lit environments. In particular, a front projection set up requires a darkened room for optimum viewing quality.

Off-screen display systems. These display systems do not utilize any projection screen. Instead, a natural medium, like simple glass or even the retina, can be used for image projection. Off-screen systems are based either on coherent or noncoherent light emission. Coherence is a property of waves that measures the ability of the waves to interfere with each other. Usually, laser light has much higher coherence than nonlaser light. VRDs (see section "Laser Displays") and 3D holographic head-up displays are examples of off-screen display systems.

Cathode Ray Tube

The CRT is a cathodoluminescent display: light is generated by exciting a luminescent material with energetic electrons. A CRT consists of a glass bulb, a cathode, an electron gun, deflection yoke, a mask, and a phosphor coating (Fig. 11.11).

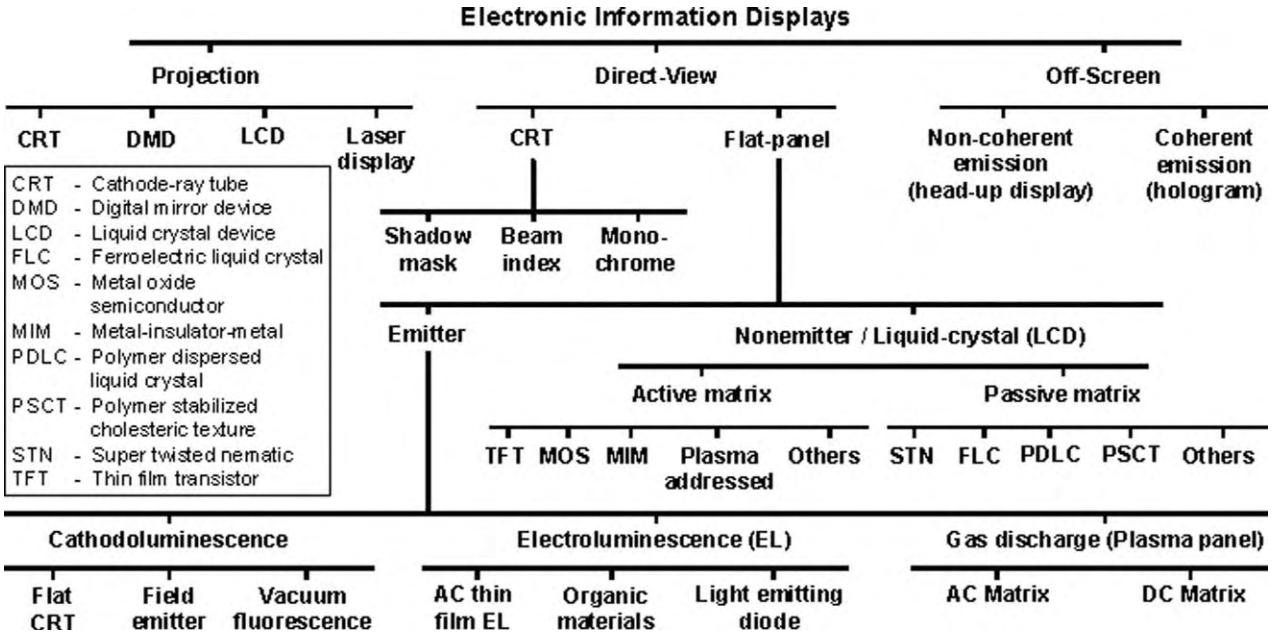


FIGURE 11.10. Classification of electronic information technologies with high information content (Theis, 1999).

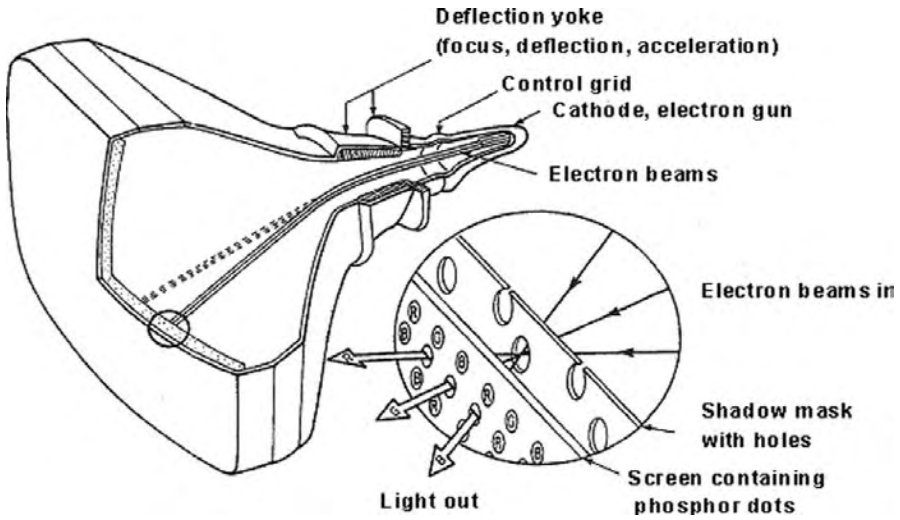


FIGURE 11.11. Major parts and components of a cathode ray tube. In this example shadow mask technology is depicted (cf. Hanson, 2003).

An electron gun located in the back of the device emits negatively charged electrons, which are attracted and accelerated by an anode that is located in front of the screen. The electron beam is diverted by an electromagnetic field, built up by the deflection coils, and thus directed toward the screen. Electrons are extracted from the cathode by thermal emission from low-surface-potential materials (typically metallic oxides), the electron beam generated at the cathode is then accelerated, deflected, and focused by a series of electrostatic lenses and deflection coils.

A screen mask is attached in front of the ground-glass plate so that the electron beam is focused and then steered on the phosphorus layer and deposited on the front surface. As a consequence, the outside areas of the beam are prevented from mistakenly hitting adjoining phosphorus dots, which would lead to a blurred representation and chromatic distortion. Display-screen masks are either dot-mask screens, Trinitrons, or slot-mask screens.

Dot-mask screen. As the name suggests, a thin metal or ceramic screen with multiple holes is used in a dot screen (Fig. 11.12 (a)). The shadow-mask technique is applied, in which three electron beams pass through the holes and focus on a single point on the tube's phosphor surface. Thus, the other display screen dots are shaded. The electron guns are arranged in the form of a triangle (delta gun), which requires a high amount of adjustment as insufficient adjustment will cause color defects. The measurement of the dot pitches of dot-type screens is taken diagonally between adjoining dots of the same color. Because of their horizontal measurement, they cannot be easily compared with the distance between the dots of a Trinitron or a slot mask screen (Blankenbach, 1999).

Slit mask (Trinitron technology). Monitors based on the Trinitron technology use an aperture grid instead of a shadow mask (Fig. 11.12 (b)). The phosphorus surface does not consist of colored dots but instead consists of a multitude of tiny vertical phosphorus wires. Their arrangement is alternating, similar to the dot mask. The electron guns are on a single line, which makes adjustment easier. Instead of a dot-type screen, the Trinitron has vertically taut wires. In contrast to the dot-type screen, the equivalent of the dot pitch, the SP of these monitors is measured by the horizontal distance between wires. Slit masks are relatively insensitive to the warmth that develops during use, because they do not bend, but merely change their length. They are very sensitive to mechanical load (i.e., vibration). Depending on the size of the monitor, either one or two horizontal holding wires are used for stabilization. These wires are recognizable (i.e., on the Trinitron they are visible as small gray lines in the top and bottom third).

Slot-mask screen. This screen technology (Fig. 11.12 (c)) also has tiny phosphor wires, which are similar to monitors based on Trinitron technology. The openings in the shadow mask are executed as slots (Precht, Meier, & Kleinlein, 1997).

CRT is a mature, well-understood technology, which is found in color and monochrome TV screens, projection TVs, vehicle displays, aircraft cockpit displays, marine instruments, VDT terminals, communication equipment, medical devices, military systems, etc. (Castellano, 1992). It offers high-information content at a low cost and is capable of displaying a large color gamut (>256 colors readily available). It is capable of high resolution and high-pixel-count displays are readily available. Direct-view displays can be made in diagonal screen sizes of up to 40 inches; projection systems using smaller tubes can be made for much larger screen viewing (Castellano). Other advantages of CRTs are an intrinsically high gray scale, good temporal dynamics, and consistent brightness for nearly all viewing angles (Candry, 2003). On the negative side, the color CRT is large and heavy, power consumption and operating voltage are high, and vibration robustness is low. Detrimental imaging characteristics include poor ambient contrast due to diffuse reflection from the phosphor surface, a propensity toward flickering, geometric distortion, and a correlation between spot size and image luminance

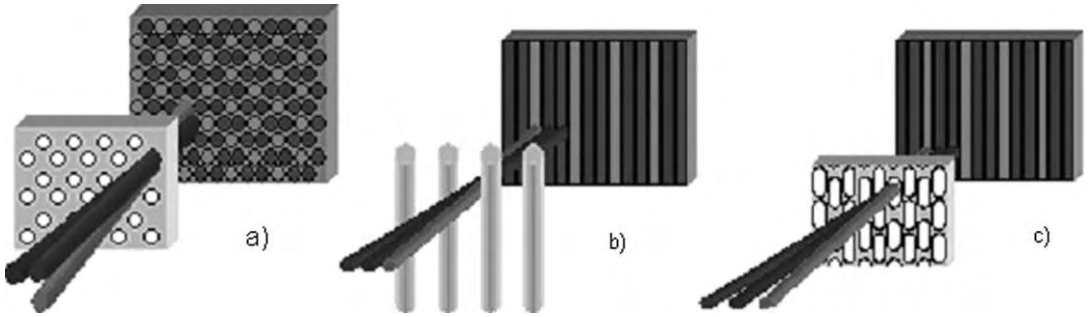


FIGURE 11.12. Dot-mask (a), slit mask (b) and slot mask (c) arrangement.

(Candry). The operation time of CRT displays is limited to about 10,000 hours of continuous operation. The main representation errors that occur in CRT displays are linearity error, convergence error, and moiré.

Linearity error. Geometric elements and letters should be represented at the same size everywhere on the screen. If displays are only adjustable vertically, problems can arise. A deflection error can cause the marginal, equispaced raster elements to disperse. Therefore, the figures' widths change constantly as they move from the middle of the screen.

Convergence error. Color representation is obtained by three electron beams and three phosphorus layers, one of each for the colors red, green, and blue. Each electron beam hits one layer. Congruent dots of the layers build a pixel that can be seen in any color, depending on the intensity of the beams. If the electron beams are not adjusted exactly, misconvergence is the consequence. "Ghostings" of the removed color arise, causing a blurred and distorted image.

Moiré. Certain color combinations and pixel arrangements can cause interference called moiré. The interference can result when a mask is deposited imprecisely, and electrons pass through the mask imprecisely. As a result, streaks are seen on the screen.

Nonemitter/Liquid-Crystal Display

Liquid Crystal Displays (LCD) have become increasingly viable in recent years and now tend to surpass CRTs. Early commercial developments of LCDs concentrated on small numeric and alphanumeric displays, which rapidly replaced LEDs and other technologies in applications such as digital watches and calculators (Bosman, 1989). Now there are even more complex displays for use in many applications, such as laptops, handheld computers, flat-panel displays (FPDs), HMDs, and miniature televisions (i.e. those in airplane seats). LCDs have the following advantages:

- Power consumption is low.
- They operate at low voltages.
- Lifetime in normal environments is very long.
- Displays may be viewed either directly in transmission or reflection or may be projected onto large screens.

LCDs consist of two glass plates with microscopic lines or grooves on their inner surfaces and a liquid crystal layer between them. Liquid crystal materials do not emit light, so external or back illumination must be provided. The physical principle is based on the anisotropic material qualities of liquid crystals. When substances are in an odd state that is somewhat like a liquid and somewhat like a solid, their molecules tend to point in the same direction, like the molecules in a solid, but can also move around to different positions, like the molecules in a liquid. This means that liquid crystals are neither a solid nor a liquid but are closer to a liquid state. LCDs operate by electrically modulating the anisotropy between optical states in order to produce visible contrast. In an electric field, liquid crystals change their alignment and therefore their translucence. If no voltage is applied, light can pass through and the pixels appear bright. When voltage is applied, the pixels become dark (Matschulat, 1999). The light to be modulated may either originate from ambient or an additional bright-light source placed behind the LCD (Theis, 1999).

The two principal flat panel technologies are the passive matrix LCD and active matrix LCD. Passive matrix addressing is used in twisted nematic (TN) LCDs. In TN LCD displays the microscopic lines of the glass plates are arranged orthogonally to each other, and the glass plates serve as polarizers (Precht et al., 1997). Their directions of translucence lie at right angles on top of one another, so that no light can pass through. Because of the fine grooves on the inner surface of the two glass panels (arranged vertically on one panel and horizontally on the other), the liquid crystal is held between them and can be encouraged to form neat spiral chains. These chains can alter the polarity of light. In the so-called nematic phase, the major axes of the crystal's molecules tend to be parallel to each other.

Nonpolarized light from background illumination can pass the polarization filter with just one plane of polarization. It is twisted about 90° along the helix and can thus pass through the second polarization layer. The display appears to be bright when there is no electric current. By applying an electric current to twisted nematics, they untwist and straighten, changing the angle of the light passing through them so that it no longer matches the angle of the top polarizing filter. Consequently, no light passes through that portion of the LCD, which becomes darker than the surrounding areas, and the pixel appears black.

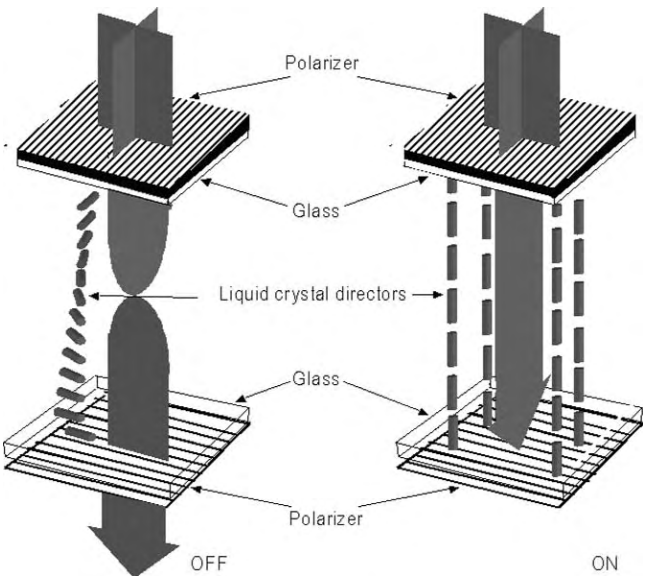


FIGURE 11.13. Principle of operation of a twisted nematic LCD.

By applying different electric currents, gray scales can be produced with LCD technology. One disadvantage with this is the relatively low contrast, but this can be improved by applying a steep electro-optic characteristic line of the liquid crystals. Low voltage is then sufficient to change the translucence, causing the liquid crystals to twist by more than 90 degrees. Such displays are called super-twisted nematic (STN), for example, double super-twisted nematic (DSTN) or triple super-twisted nematic (TSTN) (Schadt, 1996).

In passive matrix LCDs, the electric field expands over the pixel to be addressed to the entire horizontal and vertical electrodes, which results in disturbing stripes, or "ghosting." Other disadvantages are slow-response times and a prelightening of the pixels.

Active matrix (AM) addressing places an electronic switch at each pixel of an LCD; thus controlling the charging of the pixel capacitor up to the voltage corresponding to the desired gray shade, and then holding this voltage unit the next video information is written in. The available switches are thin-film transistors (TFTs), which act like diodes (Lüder, 2003). In order to display a full range of colors, each element on a LCD is assigned to a primary color by a special filter, which is deposited on the faceplate of the LCD. To guarantee high-display contrast quality, brightness and high quality of color representation, many transistors are needed. For a resolution of 1,024 × 768 pixel, 2.36 million transistors are required, one per primary color and subpixel.

The traditional representation errors caused by an incorrect deflection do not occur because of the fixed matrix, but defective transistors can cause errors, leading to permanently bright or dark pixels. These pixels are distracting, particularly when all three transistors of an elementary cell are switched to bright, which causes a white pixel to appear on the screen. Furthermore, the representation of moving pictures is restricted. The response latency of the liquid crystals is about 20 to 30 ms. Fast sequences (i.e. scrolling) are blurred.

LCD monitors with a standard video graphic array (VGA) plug have to convert the analog VGA signal back into a digital signal. If an incorrect A/D changer is used (i.e., an 18-bit-A/D changer), the color representation can be affected, although the LCD monitor is actually capable of representing true color (24 Bit).

CRTs easily represent black; the cathode simply does not emit electrons. LCDs, however, have to completely block out the light from the back light. Technically, this is impossible, and as a consequence the contrast is reduced.

When an LCD display is viewed from an angled position, it appears darker and color representation is distorted (see section "Brightness and Glare"). Recently developed technologies, such as in-plane switching (IPS), multidomain vertical alignment (MVA) and thin-film transistors (TFT) have improved the width of the viewing angle. The welcome effect of the restricted viewing angle for privacy purposes, as in the case of automated teller machines (ATMs), is worth mentioning here.

The nominal screen diagonal of a LCD is equivalent to the effective screen diagonal. In contrast, CRT nominal screen diagonals are smaller than the effective screens.

Plasma Displays

The oldest electro-optical phenomenon able to produce light is an electrical discharge in gas (Bosman, 1989). Millions of years elapsed before this effect was identified, analyzed, and mastered by humans. The first attempts to produce a matrix display panel were made in 1954. Since then, research has continued and a host of approaches have evolved (Bosman). The beauty of this technique is that, unlike front-view projection screens, one does not have to turn off the lights to see the image clearly and easily. Therefore, plasmas are excellent for video conferencing and other presentation needs (Pioneer, 2001).

In plasma technology, two glass plates are laid with their parallel thin conducting paths at right angles to one another (Precht et al., 1997). The gap between the plates is evacuated and filled with a gas mixture (see Fig. 11.14). If sufficiently high voltage is applied to the cross point of two orthogonal conducting paths, the gas ionizes and begins to shine (like a lot of small gas-discharge lamps). The inside of one glass plate is coated with a phosphorus layer, which is, according to the fundamental colors, composed of three different kinds of phosphorus. There is a wired matrix below the phosphorus-layer to trigger the PDP. The space between the glass plates is divided into gas-filled chambers. When voltage is applied to the wired matrix on the bottom of the display, the gas is transformed into a plasmatic state and emits ultraviolet radiation, causing the phosphorus to glow.

The advantage of this technology is the flat construction of large screens that perform extraordinarily well under most ambient light conditions. For example, even very bright light does not wash out the image on the screen. Another characteristic of a plasma panel is the extreme viewing angles both vertically and horizontally. With a 160°-viewing angle, people sitting to the side of the screen will still be able to see the image without losing any information (Pioneer, 2001).

Plasma screens do have disadvantages. First, they consume a large quantity of power, making this technology unsuitable for battery-operated and portable computers. Further, high voltage

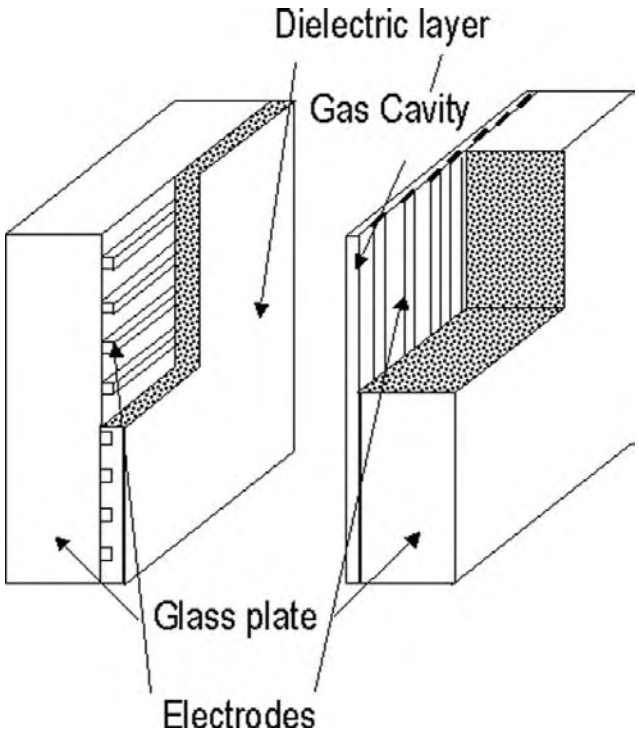


FIGURE 11.14. Major parts and components of a plasma display panel.

is required to ignite the plasma, and the phosphorus layers degrade over time. Finally, the pixels are oversized, so the user must be situated at a distance from the display.

Electroluminescent Displays

Electroluminescence (EL) refers to the generation of light by nonthermal means in response to an applied electric field that produces light (Nelson & Wullert, 1997). In early EL displays, the phosphors were used in powder form as in CRTs. Current ELDs use a thin film of phosphorescent substance (yellow-emitting ZnS: Mn), which is sandwiched between two conducting electrodes. The top electrode is transparent. One of the electrodes is coated with vertical wires and the other with horizontal wires, forming a grid. When electrical current is passed though a horizontal and a vertical wire, electrons are accelerated and pass through the phosphorous layer. Light is emitted by the excited activator atoms in the form of a pulse. The activators are transition metal or rare-earth atoms. Once excited, these centers can decay to the ground state through a radiating transition, and thus emit light with their specific emission spectrum (Budin, 2003). Electroluminescence can also be obtained from organics. Organic EL devices generally consist of a hole-transport layer (HTL) and an electron-transport layer (ETL) between electrodes. The radiating recombination can be further optimized by introducing fluorescent centers near the interface of the two layers. Such structures are particularly useful in tuning EL colors (Theis, 1999).

High-quality color presentation is the main concern with regard to ELDs. A large number of laboratories have experimented with many materials, activators, and full-color solutions (Ono, 1993). Color ELDs can be manufactured in different ways. One of the approaches, which is quite common for emitters, is that of additive-color synthesis, using three juxtaposed patterned phosphors. This approach, however, suffers from a reduced spatial-fill factor for each monochromatic emitter. The second approach to full color ELDs consists of using a single white emitting structure with patterned color filters. It results in a much simpler manufacturing process. A hybrid solution, which is being developed into a full-color commercial product at Planar (Fig. 11.15), consists in stacking and registering several plates (King, 1996). The first plate is a glass plate with the active structure on its far side, and transparent electrodes on both sides of a patterned ZnS:Mn (filtered to red) and ZnS:Tb (green) structure similar to that described above. The second glass plate has a fully transparent blue-emitting Ce:Ca thiogallate structure on top of the plate. On both plates, row electrodes are reinforced by a thin metal bus (Theis, 1999; Budin, 2003).

There are many different types of EL technology, among which the alternating current thin film electroluminescent displays (ACTFEL) are the most commonly used ELDs. Currently, ELDs are manufactured in sizes ranging from 1 to 18 inches with resolutions from 50 to 1,000 lines per inch (Rack et al., 1996). ELD qualities of note are the sharp edges of the pixels, good contrast (10:1 at 400 lux), the wide viewing angle, fast response time, durability, shock resistance, operation at high and low temperatures, and the smallest thickness and weight compared to the other flat panel displays (Budin, 2003). The bigger disadvantage of ELDs is their limited ability to be used in full-color or large-area applications (Nelson & Wullert, 1997). ELDs are primary used for medical, industrial, and instrumentation applications (Budin, 2003).

EL devices also include LEDs. LED technology is used in almost every consumer electronic product on the market. The operational principle of LEDs can be briefly described as follows: with no voltage, or reversed voltage applied across the pn junctions, an energy barrier prevents the flow of electrons and holes. When a forward bias voltage (1.5 to 2 V) is applied across the junction, the potential barrier height is reduced by allowing electrons to be injected into the p region and holes into the n region. The injected minority carriers recombine with carriers of opposite sign, resulting in the emission of photons (Castellano, 1992).

LEDs as well as some types of ELD can also be used as a backlight for LCD applications.

Cathodoluminescent Displays

The working principles of these displays are comparable to the CRT; therefore, they are also referred to as flat CRTs. One of the most successful flat CRT configurations is called vacuum fluorescent display (VFD). Vacuum fluorescent displays are broadly used in consumer electronic products and are recognizable by their green or blue-green glow, which is due to the ZnO:Zn phosphor (Nelson & Wullert, 1997). It is essentially a triode with arrays of grids and phosphor-coated anodes used for coincident

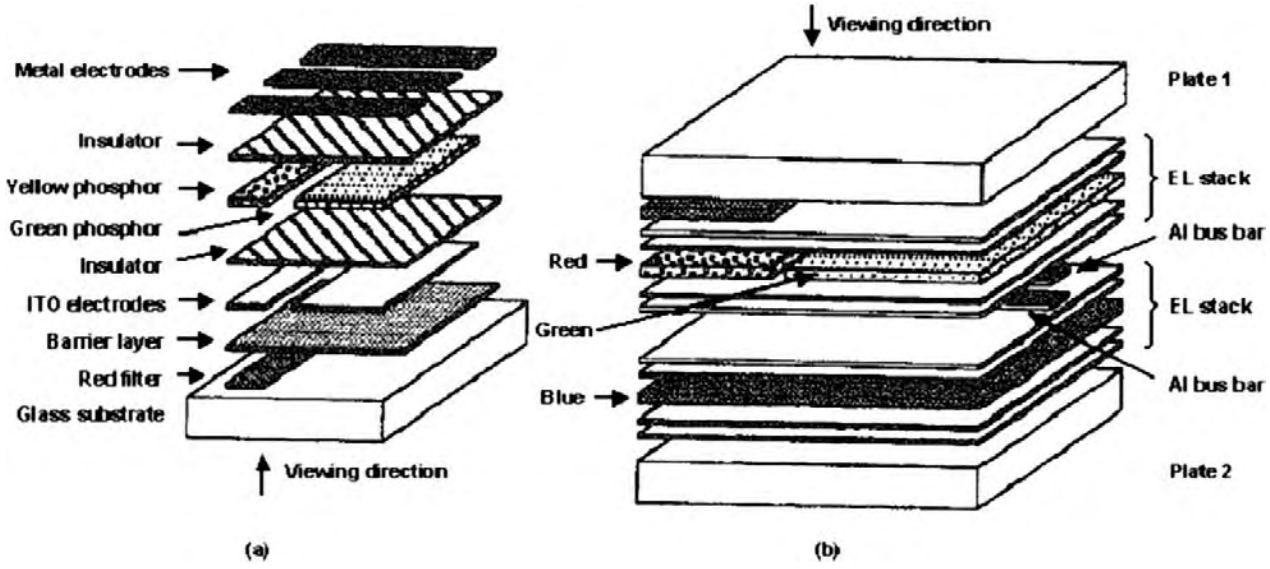


FIGURE 11.15. Schematic of (a) a Planar multicolor EL prototype, and (b) of the first full-(saturated) color dual substrate EL panel (cf. King, 1994).

addressing. The grid structure is positioned over the anode and the cathode filaments are stretched and held above the grid. The entire structure is then sealed in an evacuated cell. The cathode filament is a very fine wire that is heated to a temperature just below incandescence (about 600°C). At that temperature it remains virtually invisible but it emits electrons. The anode is supplied with a voltage of between 10 V and 50 V and at the same time the grid voltage is applied to the grid of that selected segment. The electrons, which are emitted by cathode filaments, are controlled by grids. When a grid is supplied by a positive voltage, it attracts the negative electrons whilst positive electrons are attracted when the grid is supplied with the negative voltage. The electrons collide with the phosphor-coated anode, resulting in photon emission. The VFD display can be driven in a dynamic or static system. The dynamic system is a type of multiplexing where a pulse is applied to the grid of each digit or row and to the anode of each selected digit. The static system uses a DC signal for each anode of each digit (Castellano, 1992; Nelson & Wullert, 1997).

One of the disadvantages of VFDs is that they cannot be scaled to arbitrarily large sizes because the front and back plates are only supported at the edges. Another problem is equalization of the luminous efficiencies. The blue phosphors tend to be very low in luminous efficiency and make the realization of a full-color matrix VFD difficult (Castellano).

Because the thermionic emission used in VFDs seems to be incompatible with an internal array of spacers, field emission has been proposed as a source of electrons. Internal spacers can be incorporated between emitters to allow field-emission displays (FEDs) to be thinner, lighter, and potentially much larger than VFDs (Nelson & Wullert, 1997). The basic principle of VFDs is similar to that of FEDs. In FEDs, electrons are not emitted by a cathode consisting of an oxide-coated heated filament, compared to the working principle of VFDs, but by field emission from a cold cathode.

Electrons emitted from a cathode are accelerated toward a screen, where their energy is transformed into light (Theis, 1999) (Fig. 11.16).

In real devices, efficient electron emission is obtained from a few thousand microtips per pixel formed from Mo or Si (Spindt-type, with tip radii below 1 mm for high field strength), or by replacing the microtip with planar diamond-like carbon films, which are also suitable for electrons.

The gate voltage in Spindt-type devices is about 50 V and the dielectric thickness separating gate and cathode is about 1 μm. The anode-cathode distance is 200 μm. A spacer technique is used to avoid glass bending due to atmospheric pressure (Theis, 1999; Budin, 2003).

The greatest FED challenges include finding low-voltage color phosphors, simplifying processing, and increasing efficiency (Nelson & Wullert, 1997). However, the technology is attractive since it is relatively thin and it is possible to create a Lambertian light with 180°-viewing cones and good video-response times.

Laser Displays

LASER is the acronym for Light Amplification by Stimulated Emission of Radiation. Laser light is generated in the active medium of the laser. Energy is pumped into the active medium in an appropriate form and is partially transformed into radiation energy (ILT, 2001). In contrast to thermal emitters, a laser emits a concentrated and monochromatic light with high local and temporal coherence (see Fig. 11.17).

In contrast to conventional light sources, laser light has rays that are nearly parallel with each other (it has a small divergence), is focusable down to a wave length, has a high power density, a monochromatic character (light of one wavelength), and high local and temporal coherence (the same phase). These

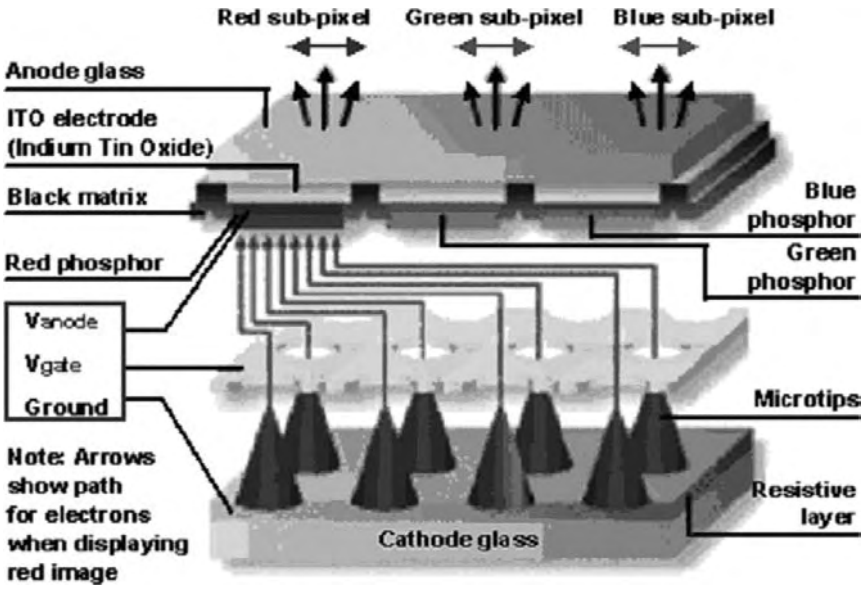


FIGURE 11.16. Principal structure of field emission displays (Theis, 1997).

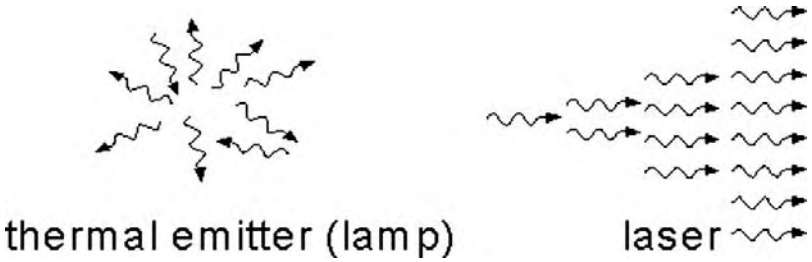


FIGURE 11.17. Dispersion of light between thermal emitters and laser light.

characteristics can be used to generate an image analogous to a conventional CRT system: the laser is directed line by line across the display area. The line diversion, horizontal and vertical, is achieved by the use of rotary or oscillating mirrors. Light output from the diode-pumped solid-state lasers is modulated according to the input signal, and the red, green, and blue light is combined. This combined light is then raster-scanned onto the screen to create an image. The laser projects the image either onto a flat area or directly onto the retina. Figure 11.18 shows the principle of a retinal-scanning laser display.

A collimated low-power laser diode is normally used as a modulated light source. The tiny laser beam is subsequently deflected in u- and v-direction by two uni-axial scanners. The horizontal scanner (u-direction) operates at several kHz. The vertical scanner frequency (v-direction) defines the image refresh rate, which must exceed the critical fusion frequency (CFF) of about 60 Hz to prevent flickering effects. Finally, a viewing optics projects the laser beam through the center of the eye pupil onto the retina. In the retinal-scanning display system the pixels are projected serially in time onto the retina. As the image refresh rate is above the temporal resolution limit of the human eye, the user does not perceive any flickering effects (von Waldkirch, 2005).

Electronic Ink and Electronic Paper

For nearly 2,000 years, ink on paper was the most popular way to display words and images. It still has many advantages compared to computer displays with regard to text readability and price. The biggest limitation of paper displays is that the printed symbols cannot be changed or removed without leaving noticeable marks. The invention of electronic ink has made it possible to overcome this disadvantage. Electronic ink displays promise to have paper-like properties. For example, electronic ink displays are viewed in reflective light, have a wide viewing angle, are thin, flexible, and relative inexpensive. Unlike paper, they are electrically writeable and erasable. A big advantage that they have over other displays is their very low power consumption, possibly extending the battery life of devices with such displays into months or even years. The principal components of elec-

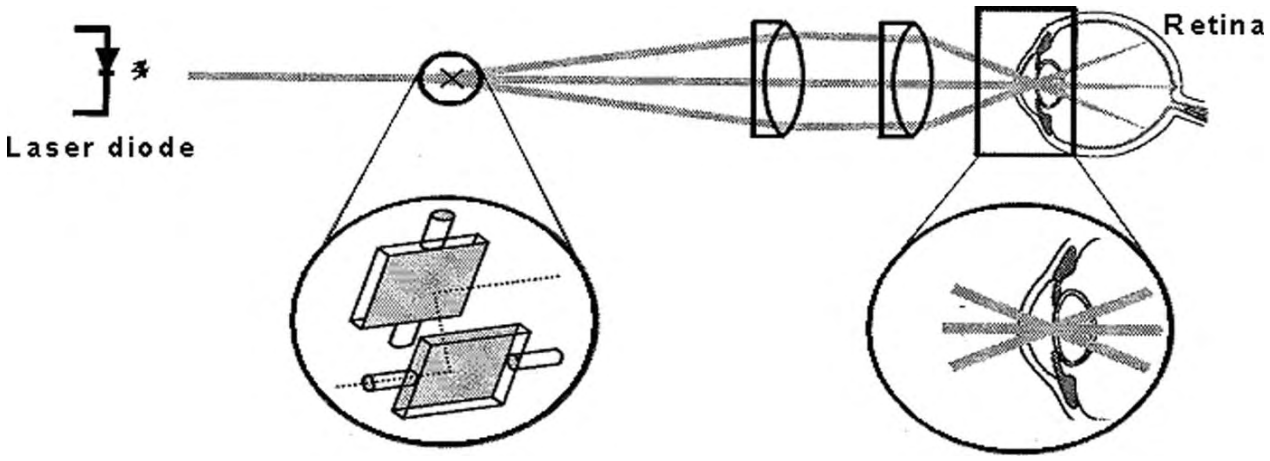


FIGURE 11.18. Schematical illustration of the RSD principle (according to Menozzi et al., 2001).

tronic ink (see E Ink Corporation, 2001) are tiny capsules, about 100 mm diameter, filled with a clear fluid. Suspended in the fluid are positively charged white particles and negatively charged black particles. The particles will move toward an inversely charged electrode. Therefore, when a top transparent electrode is negatively charged, the white particles move to the top and become visible to the user. The black particles will, at the same time, move to a positively charged bottom electrode and the microcapsule appears white to the user. Similarly, the microcapsule will appear black when an electric field of the opposite direction is applied. The microcapsules are suspended in a "carrier medium" that can be printed onto a sheet of plastic film (or onto many other surfaces). The film is again laminated to a layer of circuitry that forms a pattern of pixels that can be controlled similarly to other displays.

An invention made 20 years ago at Xerox Palo Alto Research Center forms the basis of Gyricon reusable paper, which has similar advantages to electronic ink. It consists of an array of small balls of 0.03 to 0.1 mm in diameter embedded in oil-filled pockets in transparent silicone rubber. Every ball is half white and half black and can rotate freely. Each ball has an electric charge, and when an electric field is applied to the surface of the sheet, the balls rotate showing either their black or white side and attach to the wall of the sheet. They remain there displaying the intended image as long as no new electric field is applied (Gibbs, 1998).

Unlike the first prototypes of electronic ink and electronic paper, which could only display two colors, usually black and white, photonic ink can display any color value in the spectrum. Photonics (from photon) is an area of study based on the utilization of radiant energy, such as light, for various applications. Photonic ink or p-ink, consists of planar arrays of silica microspheres—an opal film-embedded in a matrix of cross-linked polyferrocenylsilane (PFS). Photonic ink is able to display color through controlled Bragg diffraction of light. Varying the size of the spaces between the particles creates different colors. A polymer gel is filled between the stacked spheres. The gel swells when soaked in a solvent and shrinks when it dries out, changing the distance between the tiny spheres. Because the size of the spaces determines the wavelength that will be reflected, the swelling and drying of the gel results in a continuously tunable display of color across the entire spectral range visible. The amount of solvent absorbed by the gel is controlled by applying an electrical voltage. The optical response of the film to a change in solvent is less than half a second (American Chemical Society, 2003).

Applications for electronic paper are seen as a substitute for paper products such as books and newspapers, as rewriteable paper in the office, as a material for price tags and retail signs, as wall sized displays, foldable displays, and as an alternative display for PDAs and mobile phones.

VISUAL PERFORMANCE

Trouble-free usage and user acceptance of any new technology depend substantially on the quality of the visual display as a central communication unit as well as the ease with which the displays allow visual information to be processed. Thus, a careful visual evaluation is indispensable in order to assess the efficiency of visual performance and to identify existing shortcomings of the displays.

Looking back, there is quite a long history of studies concerned with the evaluation of visual displays (Dillon, 1992; Schlick, Daude, Luczak, Weck, & Springer, 1997; Luczak & Oehme, 2002; Pfendler & Schlick, in press). In order to estimate the costs and benefits of electronic displaying in terms of human performance and visual load, the main interest was to learn which factors affect visual performance to what extent. One research approach was to compare display types with respect to the efficiency of visual processing. In most cases a basic comparison of the traditional hard copy with different electronic displays was performed (Heppner, Anderson, Farstrup, & Weidenmann, 1985; Gould, Alfaro, Barnes, Finn, Grischkowsky, & Minuto, 1987). Another approach was to study the effects of

specific display factors on performance (Pfendler, Widdel, & Schlick, 2005; Sheedy, Subbaram, & Hayes, 2003; Plainis, & Murray, 2000; Ziefle, 1998). This procedure helps to distinguish different sources that account for performance shifts. Moreover, user characteristics (Bergqvist, Wolgast, Nilsson, & Voss, 1995; Kothiya & Tettey, 2001; Ziefle, 2003a) and workplace settings (Sommerich, Joines, & Psihoisos, 2001; Ziefle, 2003b) as well as individual body postures received attention (Aarås, Horgen, Bjorset, Ro, Thoresen, & Larsen, 1997; Aarås, Horgen, Bjorset, Ro, & Thoresen, 1998; Helander, Little, & Drury, 2000) in the context of visual performance. Beyond display characteristics, any visual evaluation should also consider the diverse task demands, because they can strongly influence performance outcomes. Their influence can be either positive, by compensating other suboptimal visual settings, or negative, when several negative factors accumulate, possibly leading to visual complaints.

Task Demands

Task demands are of crucial importance for visual evaluations because they represent a complex entity of different elements. The combination of (a) the task type (what the user is requested to do), (b) user characteristics (level of expertise, visual abilities or working motivation), (c) text factors (font size, line pitch), (d) display factors (i.e. contrast, refresh rates, resolution) and (e) surrounding factors (i.e. ambient lighting or time on task) has a considerable impact on performance outcomes. As these factors were found to interplay, it is necessary to examine the particular conditions and settings that were used in the different studies concerned with visual evaluations.

In order to evaluate visual displays, simple detection tasks, memory and recognition tasks, visual search tasks, and (proof) reading were used. Overall, two basic task forms can be differentiated between in this context: tasks in which a semantic context is present (i.e., (proof) reading) and tasks with no semantic context (i.e., visual search). This distinction has implications with respect to the outcomes' ecological validity and generalizability as well as to the sensitivity with which shortcomings of visual displays are reflected. Tasks that have a semantic context have the general advantage of simulating what users mostly do when using displays, thus the evaluation process is ecologically valid. However, these tasks were found not to be very sensitive to visual degradation effects (Stone, Clark, & Slater, 1980; Ziefle 1998), because of two main reasons. First, the encoding and processing of text material represents behavior gained through intensive training: representing a top-down process, comprehension guides the reader through the text and possibly masks degradation effects. The second objection is concerned with reading strategies involving different combinations of cognitive and visual processes: when proofreading, for example, participants may read for comprehension or scan for unfamiliar letter clusters and word shapes. Thus, it is possible that performance outcomes do not reflect the degradation properly even though the display quality is suboptimal: deteriorations might then be overlooked, especially in short test periods. Even though ecological validity is lower, visual search and detection tasks showed a higher sensitivity to visual degradation effects, because the visual encoding process predominately relies on visual properties of the display (bottom-up process) without being masked by compensating cognitive strategies.

However, text factors were also found to considerably affect performance: among those, display size, the amount of information to be processed at a time, as well as font size (Duncan & Humphreys, 1989; Oehme, Weidenmaier, Schmidt, & Luczak, 2001, Ziefle, Oehme, & Luczak, 2005) were revealed to be crucial in this context. The smaller the font size and the larger and denser the amount of (text) material to be processed, the stronger the performance decrements were.

Whenever a critical visual evaluation procedure is needed that reflects effects of visual displays on performance, a two-step procedure is to be recommended. As a first step, a benchmark procedure is advised. This procedure includes young and well-sighted participants and a task that is visually rather than cognitively strenuous. The second step includes a broadened scope. Older users (as they represent a major part of the workforce) and tasks with different visual and cognitive demands have to be addressed. In addition, extended periods of on-screen viewing are to be examined in order to realistically assess the long-term effects of computer displays.

Measures Used for Visual Evaluations

The measures used to quantify the effects of visual display quality also differ regarding their sensitivity to visual degradation. Apart from effectiveness and efficiency of performance, judgments of visual strain symptoms and the presence of visual complaints were also assessed.

Among performance measures, global and local parameters can be distinguished. Global performance measures are the speed and accuracy of task performance; they were widely used across visual evaluation studies. Reading and search times reflect the basic velocity of the encoding and visual processing. However, system speed can only then be meaningfully interpreted when the accuracy of task completion is also taken into account. In this context, the speed-accuracy trade off has a considerable impact. Whenever visual conditions, for example in computer displays, are suboptimal (under low resolution or contrast conditions), the speed and accuracy of information processing cannot be kept at a constantly high level simultaneously. Rather, one of the two components deteriorates while the other is kept constant. When both components are considered, the overall costs for the information processing can be assessed. Nevertheless, reliable, global measures do not provide an understanding of the processes that cause the performance decrement. Therefore, the oculomotor behavior was consulted in order to gain deeper insights into the nature of the deteriorated encoding process (Owens, & Wolf-Kelley, 1987; Iwasaki & Kurimoto, 1988; Jaschinski, Bonacker, & Alshuth, 1996; Best, Littleton, Gramopadhye, & Tyrell, 1996). The spatial and temporal characteristics of saccades were predominantly analyzed. Under visual degradation conditions, more saccades are executed, smaller in size, and accompanied by increased fixation times (Ziefle, 1998; Ziefle 2001a, 2001b). Also, accommodation states, pupil size, vergence efforts, and visual scan path complexity (Schlick, Winkelholz, Motz, & Luczak, 2006) were taken as measures to quantify the effects of display quality. The higher effort

required for the visual system observed when reading electronic texts is assumed to form the physiological basis of visual fatiguing (Wilkins, Nimmo-Smith, Tait, McManus, Della Sala, Tilley, et al., 1984). Beyond oculomotor measures, stress indicators such as heart-rate variability and eye blinks were used to determine the effects of display quality (Oehme, Wiedenmaier, Schmidt, & Luczak, 2003).

Another approach to quantifying effects of display quality is to collect user judgments with respect to what is called visual fatiguing, visual stress, strain, visual load, or discomfort. Visual fatiguing is considered as a complex subjective measure based on the awareness of several symptoms: burning, dry, aching, and watering eyes, difficulties in reading (text becomes blurred and fades away) as well as increased blinking and eye pressure (Stone et al., 1980; Hung, Ciuffreda, & Semmlow, 1986). User judgments represent an important aspect of display quality, as perceived visual comfort is the most direct source of users' satisfaction. Even though ratings are easily to obtain, their validity is not without controversy. First, users differ considerably with respect to their responsiveness to visual strain symptoms. Furthermore, the sensitivity to visual stress is not constant, but changes with the time spent on the task and age (Wolf & Schaffra, 1964). Second, factors like fear of failing and the misinterpretation of one's own performance may contaminate judgments and make it necessary to prove that ratings match performance outcomes. Third, the emergence of visual fatiguing does not necessarily follow the same time course as performance shifts. Visual fatiguing is therefore not necessarily accompanied by performance decrements and vice versa (Yeh & Wickens, 1984; Howarth & Istance, 1985). In order to obtain a valid picture, it is therefore advisable to include both subjective and objective measurements.

Effects of Specific Display Factors on Visual Performance

As a first factor, effects of display resolution (see section "Resolution") are considered. Technically speaking, display resolution is defined as the number of pixels per inch out of which objects and letters are displayed on the screen surface. The psychophysical correlate of display resolution, and this is of visual ergonomic interest, is the sharpness of contours and the clarity with which objects can be identified. High-display resolutions allow more objects to be displayed on a given screen space, as object sizes decrease with increasing resolutions. Even if this may be advantageous when screen space restrictions are considered, it might be counter-productive: the benefit of higher resolutions in terms of contour sharpness may be negated by the smaller object sizes, and this has to be counterbalanced. The central question is whether higher resolutions lead to quantifiable improvements. The relevant literature shows that this is indeed the case (Gould et al., 1987; Miyao, Hacisalihzade, Allen, & Stark,1989; Ziefle, 1998). Due to unequal technical standards over time, resolution levels differ considerably between the studies (40–90 dpi). In conclusion, it can be said that up to 90 dpi performance, the higher resolution, is better. Young adults' search performance was increased by 20% in the 90 dpi condition compared to the 60 dpi condition. This greater effectiveness was caused by a more efficient oculomotor control: fixation times decreased by 11% and, in addition, less saccades (5%) were executed for processing the visual information. Furthermore, performance in the low-resolution condition was not only found to be interrelated with the emergence and strength of visual fatiguing symptoms, but also, the probability of fatiguing symptoms was significantly higher when two suboptimal viewing conditions were coincident: the longer participants worked in the low-resolution condition the stronger performance decrements were (Ziefle, 1998).

A second factor to be examined extensively was intermittent light stimulation, which is characteristically present in CRT screens, referred to as the "refresh rate" (see section "Refresh Rates"). The perceptual component here is a flicker sensation, which predominately occurs when low refresh rates are present. Reading on a screen with low refresh rates (50 Hz) is extremely hard work for users and leads to considerable eyestrain symptoms even after short reading periods. With increasing refresh rates (>70 Hz) the flicker sensation decreases, but it should be noted, that physically, the intermittent stimulation is nevertheless still present and may affect performance. Many ergonomic studies have concerned themselves with screen flicker and found that it is a major source of performance decrements and visual fatiguing, involving the emergence of migraines (Boschmann & Roufs, 1994; Jaschinski et al., 1996; Küller & Laike, 1998; Lindner & Kropf, 1993). The nature of the flicker effect is referred to as "a basic disturbance of eye movement control" caused by the intermittent light (Kennedy & Murray, 1993; Baccino, 1999). Regarding the impact of refresh rates on performance, it was shown that performance with the 100 Hz screen was 14% better compared to 50 Hz, and ocular efficiency was 16% better (fixation times and number of fixations per line). Apart from that, effects of the time spent on the task were found (comparing the beginning and the end of the task completion). Already in a test period of only 30 minutes, visual and ocular performance decreased by 8% (Ziefle, 2001a). However, effects of refresh rates on performance do not follow a linear ("the higher the better"), but rather a curvelinear relationship: when a 140 Hz screen was compared to a screen driven with 100 Hz, search performance and ocular efficiency were found to deteriorate in the 140 Hz condition (by 5–10% across measures). This confirms that the visual system is sensitive to high-frequent intermittent light, even though users are no longer able to detect flicker in this high refresh-rate condition. Overall, we can say in summary that refresh rates of 90–100 Hz in CRTs facilitate reasonably good performance.

Third, effects of luminance (see section "Brightness and Glare") and contrast (see section "Resolution") were also reported to play a major role for visual performance when computer displays are used (Plainis & Murray, 2000; van Schaik & Ling, 2001; Ziefle, Gröger, & Sommer, 2003; Oetjen, Ziefle, & Gröger, 2005). However, results are not easy to integrate. The contrast itself is a complex factor, but it is also escorted by several other lighting characteristics, which are highly interdependent and affect visual performance outcomes separately as well as in combination. The contrast, always a ratio out of two luminance levels (background and object) was proven to markedly affect visual performance. Low-contrast ratios lead to performance decrements and oculomotor efficiency in a range of 10–20%. But contrast ratios do not specify which absolute luminance levels constitute the respective contrast and—depending

on which contrast definition is used—they mostly do not differentiate between whether information is displayed in negative or positive polarity (bright letters on dark vs. dark letters on a bright background). In addition, ambient lighting considerably interferes with the display contrast. Thus, knowing the contrast level without detailing polarity, absolute luminance levels and the level of ambient lighting does not allow the interpretation of performance outcomes. Generally, it was found that positive polarity displays result in a better visual performance than negative ones (Bauer & Cavonious, 1980). The positive effect of positive polarity occurs due to the (simple) fact that light is essential for visual encoding (the absolute light level is higher with positive displays), and because it is less demanding when the computer display has the same polarity as hard copy (especially because both media are often used simultaneously in the same work setting), and does not require the user to alternately readapt. Ambient lighting strongly reduces the contrast levels on the screen, because the two lighting sources interfere. Thus, it is advantageous for performance if illumination in the room is sparse (Kokoschka & Haubner, 1986). Regarding display luminance it is recommended that luminance levels should not be too bright, preventing interfering glare, especially when ambient lighting is low and the probability of glare rises (Schenkmann, Fukunda, & Persson, 1999).

The importance of contrast and luminance has received renewed interest in recent studies (Hollands, Cassidy, McFadden, & Boothby, 2001; Hollands, Parker, McFadden, & Boothby, 2002; Gröger, Ziefle, & Sommer, 2003; Ziefle et al., 2003; Oetjen & Ziefle, 2004). Among the visual factors, TFT screens (see section "Nonemitter/Liquid-Crystal Display") have the basic advantage of being flicker free. However, they also have one major disadvantage: the displayed information is "perfectly" visible if users work in front of the TFT screen. Whenever this "optimal" position is not present, visibility is distinctly worse. This specific property of TFT screens is called "anisotropy." A display is called anisotropic if it shows a deviation of more than 10% of its luminance subject to the target location or viewing angle (ISO 13406-2 2001). The nature of anisotropy is such that photometric measures (contrast and luminance) are not constant over the screen surface, but rather decrease with increasing viewing angle. In order to quantify the change in photometric measures at different viewing angles, a measurement set up was developed that made it possible to exactly correlate photometric measures and visual performance (Gröger et al., 2003; Ziefle et al., 2003). The screen was virtually cut into 63 (9×7) fields. The luminance of bright/dark areas was individually measured by a photometer and contrasts were determined. Then different measuring positions were adopted: first, the "standard view" was applied, commonly used by the industry. The photometer was set in front of the screen and displaced gradually from field to field, with the photometer always set at right angles to the screen (Fig. 11.19 left).

From an ergonomic point of view, this procedure (as shown in Fig. 11.19, left) is highly artificial, as users do not displace themselves, but rather change their view: viewing angles change remarkably depending on where users are looking. This is entirely disregarded in this measurement procedure. In order to simulate real viewing conditions, the "user view" (Fig. 11.19, center) as well as the "bystander view" (Fig. 11.19, right) were realized. For the user view, the photometer was positioned centrally in front of the screen. As the position of the photometer did not change, viewing angles increased with distance from the center of the screen, thereby emulating the user's head movements when looking toward the screen edges. For the "bystander

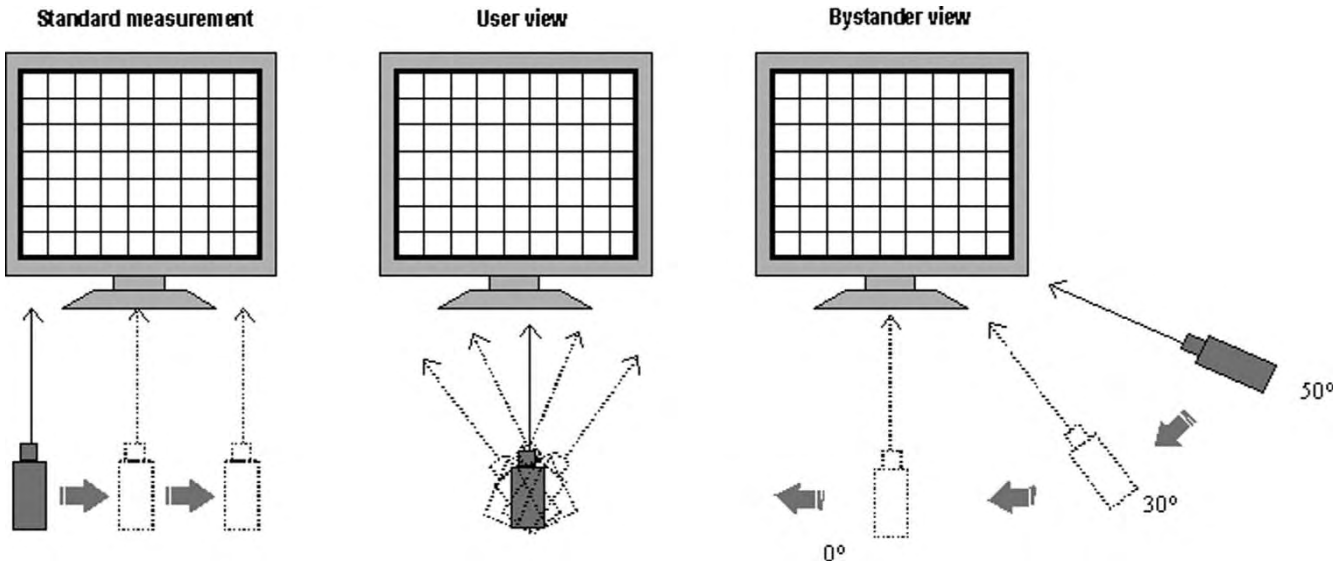


FIGURE 11.19. Quantifying anisotropy in TFT screens: left: "standard view" with the photometer displaced at right angles; centre: "user view" with the photometer emulating the user's head movements; right: "bystander view" with the photometer positioned off-axis (Ziefle et al., 2003).

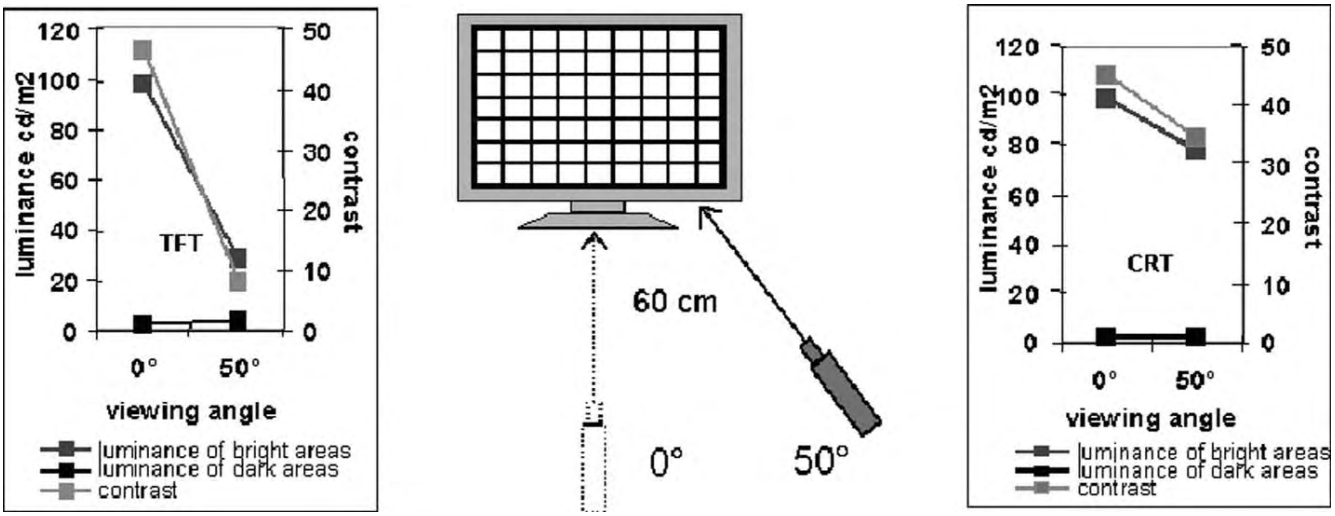


FIGURE 11.20. Photometric measures in the "bystander view" for a TFT (left side) display and a CRT (right sight) display (Oetjen & Ziefle, 2004)

view," the photometer was set to a central point of the display and turned to the different measuring fields (30° and 50°). For extended-viewing conditions, the photometer was set off axis, and its view pointed to the screen from the side (left and right side, respectively).

In Fig. 11.20 it becomes evident that photometric measures change dramatically as a function of viewing angle in TFT Displays (Figure 11.20, left), and to a much lesser extent in CRTs (Fig., 11.20, right).

Performance outcomes showed that anisotropic effects have to be taken seriously. Visual performance with TFT screens considerably deteriorates by about 10% for young adults when they had to look at the screen off axis. Anisotropy plays a role in many real-life work settings: (air, rail) traffic controlling environments or stock exchanges use several TFT screens simultaneously (placed parallel to and/or upon another) that have to be surveyed by one operator. Another example is the schooling and training context where it is quite usual for several users to work with only one screen. Thus, from an ergonomic point of view, TFT's anisotropy must be regarded as a visual limitation of the TFT Display technology, at least when a fast and accurate visual detection performance is of importance.

Comparison of Different Display Types

During the last 30 years, since the first evaluation studies of different displays were published, a huge number of studies have dealt with the fundamental question as to which display type assures the highest reading comfort and the best visual performance. Typically, and this reflects the chronological development, hard copy, CRT, and TFT displays were compared with respect to visual performance. As electronic display quality has improved continuously over time, the technical standards on which the evaluations were based, differ greatly. This is a factor that should be borne in mind. Recent studies include new developments (augmented reality, see section "Augmented Reality") in visual-display technology.

However, independently of the time and technical standard of computer displays, the one and only display that outperformed all others with respect to visual performance and comfort is the traditional hard copy. Its development covers more than 200 years (since the invention of the industrial production of hard copy in France) and has been continuously amended with respect to readability and visibility, by the expertise of typesetters and typographers. Thus, hard copy can be regarded as an outstandingly suitable display with regard to visual ergonomic demands. It provides high contrast and resolution without disturbance by glare, screen reflections, or flicker. Accordingly, the majority of studies show that performance in CRT screens is significantly lower compared to paper (Gould et al., 1987; Heppner et al., 1985; Ziefle, 1998). Hardcopy was also found to outperform modern TFT displays (Ziefle, 2001a).

As the private and public need for electronically displayed information continuously increases the evaluation should focus on the quality of different electronic displays. Here, the classical comparison of CRT and TFT displays is of central interest. The CRT, the hitherto widespread screen type, seems to be being phased out due to its specific lighting characteristics (flicker). The development of TFT technology was therefore highly welcome: TFT screens, lightweight and flat, are flicker free and display information at much higher levels of luminance and contrast.

At first sight, the outcomes of studies that compare CRTs and TFTs reveal an inconsistent picture: There are studies in which TFT displays led to a higher visual performance compared to CRT screens (Menozzi, Lang, Näpflin, Zeller, & Krueger, 2001; Ziefle 2001a,b), but there are also studies in which CRT displays outperform TFT screens (Oetjen & Ziefle, 2004; Oetjen et al., 2005; Oetjen & Ziefle, in press). This inconsistency can be resolved when taking the experimental purposes and evaluation settings into account. Whenever the impact of screen flicker was of main interest, the methodological focus was to compare both displays with respect to refresh rates (present in the CRT and absent in the TFT). This procedure ruled out any other differences between both displays that may confound or compensate flicker effects (anisotropy). Thus, participants were usually seated in front of the screen and had to work on tasks that were displayed centrally (otherwise, anisotropic effects would have been mixed up). In these cases, visual performance clearly favors the TFT and shows that screen flicker in CRTs is disadvantageous. In addition, it was found that the relative benefit TFT technology is disproportionately higher for older users: while young adults (20–25 years) showed performance superiority of 10% for the TFT compared to the CRT (100 Hz), the benefit from the TFT was 16% and 27%, respectively, when older users (40–65 years) were examined (Ziefle, 2001b). Note, however, that the older adults generally showed nearly 40% lower visual performance.

However, the benefit of the TFT thus determined has not yet taken anisotropic effects into account. As soon as anisotropy is considered, the picture changes. Studies concerned with the two display types from the anisotropy perspective showed that the TFT superiority over the CRT disappears when extended viewing angles were taken into account (Gröger et al., 2003; Hollands, et al., 2001, 2002; Oetjen & Ziefle, 2004; Oetjen & Ziefle, in press). When considering all screen positions, visual performance decreased by 8% when a TFT was used instead of a CRT screen. When a central view was applied, detection times were 14% faster than when viewing 50° off axis. A further aggravating factor was font size. Deterioration when detecting small targets (1.5 mm vs. 2.4 mm) rose to almost 30% with the TFT, in contrast to "only" 20% with the CRT. Whenever all sub-optimal factors (small font, TFT screen, off-axis viewing) occurred simultaneously, performance decrements were found to mount up to as much as 38% (Oetjen & Ziefle, 2004).

STANDARDS, RULES AND LEGAL REGULATIONS AND HEALTH PROTECTION ASPECTS

There are a large number of national and international standards, regulations and seals of approval that regulate diverse aspects of the design and use of visual displays. Among the areas covered are ergonomics, emissions, energy consumption, electrical safety, and documentation. Due to very fast technological development, only selected standards are listed in Table 11.1. Up-to-date information can be found on the websites listed in Table 11.2.

DISPLAYS FOR SELECTED APPLICATIONS

Virtual Reality

Immersive virtual reality (VR) is a technology that enables users to "enter into" computer-generated 3D environments and in-

TABLE 11.1. A Selected List of National and International Standards, Regulations and Seals of Approval Regulating Design and Use of Visual Displays

Ergonomics	
Image quality	DIN EN 29241-3, ISO 13406-2, TCO'99, GS-Mark, Ergonomics Approved Mark
Reflection characteristics	DIN EN ISO 9241-7, ISO 13406-2, TCO'99, GS-Mark, Ergonomics Approved Mark
Color requirements	DIN EN ISO 9241-8, ISO 13406-2, TCO'99, GS-Mark, Ergonomics Approved Mark
Brightness and contrast adjustable	European VDU directive 90/270 EEC, German ordinance for work with visual display units, TCO'99, GS-Mark, Ergonomics Approved Mark
Tilt and swivel	TCO'99, GS-Mark, German ordinance for work with visual display units
Gloss of housing	GS-Mark, German ordinance for work with visual display units
Emissions	
Noise	European VDU directive 90/270 EEC, ISO 7779 (ISO 9296) German ordinance for work with visual display units, TCO'99, GS-Mark
Electrostatic potential, electrical and magnetic fields	PrEN 50279, TCO'99, Ergonomics Approved Mark
X-ray radiation	TCO'99
Energy consumption	
	EPA Energy Star, TCO'99, VESA DPMS
Electrical safety	
	TCO'99, GS-Mark
Documentation	
Technical documentation and user manual	German Equipment Safety Law, GS-Mark

TABLE 11.2. Websites that Provide Information about National and International Standards, Regulations and Seals of Approval for Visual Displays

Websites	
European directives	European directives regarding health and safety are available via the website of the European Agency for Safety and Health at Work. Internet: http://europe.osha.eu.int/legislation/
EPA	The US Environmental Protection Agency (EPA) promotes the manufacturing and marketing of energy efficient office automation equipment with its Energy Star Program. Internet: http://www.energystar.gov/
National laws and ordinances	National laws, directives and regulations regarding health and safety for many European and some other countries are available via the website of the European Agency for Safety and Health at Work. Internet: http://europe.osha.eu.int/legislation/
ISO	The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies from 140 countries. Internet: http://www.iso.ch
EN	European standards are available via the website of the European Agency for Safety and Health at Work. Internet: http://europe.osha.eu.int/legislation/standards
DIN	DIN is the German Institute for Standardization Internet: http://www.din.de
TCO	TCO (The Swedish Confederation of Professional Employees) developed requirements for PCs. The TCO'99 label specifies ergonomic, ecological, energy consumption and emission requirements. Internet: http://www.tco.se
VESA	The Video Electronics Standards Association (VESA) creates standards for transmissions between computers and video monitors that signal inactivity. Internet: http://www.vesa.org
GS-Mark	The German safety approval mark shows conformity with the German Equipment Safety Law and signals that the product, as well as the user manual and production process, has been tested by an authorized institution such as TÜV Rheinland. Internet: http://www.tuv.com
Ergonomics Approved Mark	The Ergonomics Approved Mark demonstrates that a Visual Display Terminal complies with numerous ergonomic standards. The test mark is devised by TÜV Rheinland. Internet: http://www.tuv.com

teract with them. VR technology involves the additional monitoring of body movements using tracking devices, enabling intuitive participation with and within the virtual world. Additional tracked peripheral devices permit virtual navigation, pick-and-place manipulation of virtual objects (Schlick, Reuth, & Luczak, 2000), interaction with humanoids and avators by using data gloves, space joysticks, or 3D-tracked balls (Holmes, 2003).

A 3D-view is generated using different concepts and technologies (see sections "Human-Related Display Functionality and other Criteria"). The HMD is a commonly used display device. For VR, a closed-view HMD in non-see-through mode is used (see section "Augmented Reality"). Visual displays, especially HMDs, have decreased substantially in weight since the first invention of an immersive head-worn display, but are still hindered by cumbersome designs, obstructive tethers, suboptimal resolution, and an insufficient field of view. Recent advantages in wearable computer displays, which can incorporate miniature LCDs directly into conventional eyeglasses or helmets, should simplify ergonomic design and further reduce weight (Lieberman, 1999). Most of the advanced closed-view HMDs have adjustable inter pupillary distance (IPD) in order to avoid mismatches in depth perception. They provide a horizontal field of view of 30 to 35 degrees per eye and a resolution of at least 800 600, and therefore outperform predecessor systems (Stanney & Zyda, 2002). Large images of VR scenes can be reproduced by a projector on the front or back of one or more screens. Stereoscopic images for both eyes are projected either alternately or with different polarization at the same time (see section "Human-Related Display Functionality and Other Criteria").

The "CAVE" (Computer Animated Virtual Environment) is a further development in projection technology. It consists of a cube with several panels onto which the images are projected from behind. Depending on the construction they are C3 (two walls and the floor), C4, C5, or C6, respectively. A CAVE provides space for small groups, but can track and optimize the stereoscopic view for only one person. The others perceive distortions, especially on corners and edges.

Augmented Reality

Augmented Reality (AR) characterizes the visual fusion of 3D virtual objects into a 3D real environment in real time. Unlike virtual reality or environments, AR supplements reality, rather than completely replacing it (Azuma, 2001). AR can be used in many applications, such as production (Schlick et al., 1997), assembly and service, medical (Park, Schmidt, & Luczak, 2005), architecture, entertainment and edutainment, military training, design, robotics, and telerobotics.

One approach to overlaying the real world with virtual information is to use an HMD. Superimposition can occur in two ways: using an HMD with see-through mode (optical-see-through) and an HMD with non-see-through mode, called video-see-through (feed-through). The HMD in the non-see-through mode, which is also used in virtual reality, visually isolates the user completely from the surrounding environment, and the system must use video cameras to obtain a view of the real world. The optical see-through HMD eliminates the video channel; the user is directly looking at the real scene. Instead,

the merging of real world and virtual augmentation is performed optically in front of the user using a mini projector and half-silvered mirror. Both systems have advantages and disadvantages in usability and technology. In an optical see-through system there is a time lag between the real-world information and the virtual information that is blended into the field of view, which is caused by the computing time for image generation. In addition, the calibration of such a system is rather complicated. In video-see-through systems the quality of the video depends on the technology of the cameras and the display. There are several factors (i.e. limited refresh rate, displacement of the cameras from eye level, and time delay) that have an adverse effect on human perception and hand-eye coordination (Biocca & Rolland, 1998; Oehme et al., 2001; Luczak, Park, Balazs, Wiedenmaier, & Schmidt, 2003; Park et al., 2005; Ziefle et al., 2005). Another disadvantage is the severe loss of information because the perception of the real-work environment is limited to the maximum resolution and FOV of the displays and cameras.

Another approach to overlaying the visual information is to use a VRD. In contrast to the screen-based HMDs just described, the VRD reaches the retina directly with a single stream of pixels, thus guaranteeing a clear, sharp projection of different kinds of information (Microvision, 2000). Because of the higher laser-light intensity, the half-silvered mirrors commonly used for see-through HMDs can be optimized to a maximum translucence, improving the see-through quality. Furthermore, the display's maximum resolution is no longer determined by the tolerances of manufacturing of the display pixels, but by the control logic and quality of the deflection mirror.

A clip-on display is a combination of the optical and video-see-through modus. A tiny LCD screen with video-based real-time images can be clipped onto eyeglasses or safety glasses. The user can easily move back and forth between the images on the monitor and the real world beyond it.

Mobile Phones and Handheld Devices

Mobile phones and handheld electronic devices represent one of the fastest-growing technological fields ever. Already in 1999, 500 million mobile phones were sold worldwide, and it is forecasted that about 2.6 billion mobile devices will be in use by the end of the decade. Mobile communication technologies, such as the Internet, universal mobile telecommunications systems (UMTS), wireless local area networks (WLAN) and wireless application services, have changed and will change social, economic, and communicative pathways in private and business sectors still more. Mobile computing has already expanded into many service fields, including sales, health care, administration, and journalism. Different mobile or handheld devices can be found on the market: mobile phones, smart phones, communicators, and PDAs. The main reason for their topicality is that the devices provide on-the-go lookup and entry of information, quick communication and instant messaging (Weiss, 2002). Handheld devices may differ with respect to physical dimensions, (color) resolution, contrast, luminance, and touch/stylus sensitivity, but they are all characterized by small screen size, which has a considerable impact on information processing. Displays of mobile phones usually vary between 3 and 4 cm, and have fewer than 6 lines with up to 20 characters per line. PDAs and smart phones provide somewhat more display space (about 5×7 cm), which can result in up to 20 lines of 20 characters each.

The limited screen space is very problematic for providing optimized information access and the question of how to "best" present the information on the small display is challenging. At first glance, the challenge seems to be mainly related to visibility. If so, visually ergonomic principles are to be considered predominately: in order to provide fast and accurate information access, objects and letters should be big enough, text lines should not be too close together and information density should be low. This is especially important for older adults, who often have problems with sight (Brodie, Chattratichart, Perry, & Scane, 2003; Omori, Watanabe, Takai, & Miyao, 2002). However, visibility concerns are not the only point at issue. There is also the cognitive aspect of information visualization, the requirement that the presentation of information should help users to orientate themselves properly. Disorientation in handheld devices' menus is a rather frequent problem (Ziefle & Bay, 2005, 2006). Users have to navigate through a complex menu of functions, which is mostly hidden from sight, as the small window allows only few functions or small text fragments to be displayed at a time. Considering visual and cognitive demands concurrently, two alternatives can be contrasted: one alternative is to display only a little information per screen at a time, which helps avoid visibility problems due to high information density. The other alternative is to display as much information per screen as possible; this allows users to have maximal foresight (cognitive preview) of other functions on the menu, which should benefit information access from a cognitive point of view and minimize disorientation. Apparently, a sensitive cutoff between visual and cognitive impacts has to be met and it is to be determined whether the impact of the cognitive preview, or rather the impact of visual density is decisive for efficient information access. This was examined in a recent study (Bay & Ziefle, 2004). Young adults processed tasks on a simulated mobile phone where one, three, or seven menu items were presented at a time. Results corroborated the meaningful impact of information presentation on small screens for efficiency when using the device: intermediate foresight (three functions) was found to lead to the best performance. If only one menu item was shown at a time—as is the case in a number of devices on the market—the users needed 40% more steps to process tasks than when three items were shown, confirming the cognitive facet to be crucial. But also when information density was high (seven functions), performance declined by more than 30% compared to the presentation of three menu functions per screen, confirming the visual facet as also playing an important role.

In order to accommodate the problem of displaying a lot of information on a small screen, several other techniques were proposed. One technique is rapid serial visual presentation (RSVP), which is based on the idea of presenting information temporally instead of spatially (Rahman & Muter, 1999; Goldstein, Öqvist, Bayat, Ljungstrand, & Björk, 2001). With RSVP, one or more words are presented at a time at a fixed location on the screen and users have to integrate the single text fragments bit by bit, by scrolling the text forwards and (if necessary) backwards. A similar technique is the times square method (TSM)

also known as "leading." Here, the information is not static, but moves across the screen (word by word or sentence by sentence), with the text scrolling autonomously from left to right. Even if trained readers may reach a reasonable level of efficiency in both presentation modes (RSVP and TSM), users report disliking both presentation forms. The low acceptance may be due to a high cognitive and visual demand imposed by the information presentation: either, memory load is high, as it is difficult for users not to lose the plot while integrating the words and sentences, which are displayed one after another (RSVP). Or, visual and attentional demands are high as it is crucial to catch the text contents in moving sentences. Accordingly, when users fail to catch the contents on the first attempt, they have to wait until the scrolling information appears again (TSM).

References

Aarås, A., Fostervold, K., Ro, O., Thoresen, M., & Larsen, S. (1997). Postural load during VDU work: a comparison between various work postures. *Ergonomics, 40*(11), 1255–1268.

Aarås, A., Horgen, G., Bjorset, H.-H., Ro, O., & Thoresen, M. (1998). Musculosceletal, visual and psychosocial stress in VDU operators before and after multidisciplinary ergonomic interventions. *Applied Ergonomics, 29*(5), 335–354.

American Chemical Society (2003). *Multicolored Ink*. Retrieved October 1, 2005, from http://pubs.acs.org/cen/topstory/8112/8112notw8.html

Aye, T., Yu, K., Kostrzewski, A., Savant, G., & Jannson, J. (1998). Line-image-scanning head-mounted display. *In SID International Symposium (SID '98), Digest of Technical Papers*, *29*, San Jose, CA, USA, 564–567.

Azuma, R. (1997). A survey of augmented reality. *Presence, 6*(4), 355–385.

Azuma, R. T. (2001). Augmented Reality: Approaches and technical challenges. In W. Barfield & T. Caudell (Eds.), *Fundamentals of wearable computers and augmented reality* (pp. 27–63), Mahwah, NJ: LEA.

Baccino, T. (1999). Exploring the flicker effect: The influence of in-flight pulsations on saccadic control. *Ophthalmologic and Physiological Optics, 19*(3), 266–273.

Bass, M. (1995). *Handbook of optics*, Vol. 1. New York: McGraw-Hill.

Bauer, D., & Cavonius, C. R. (1980). Improving the legibility of visual display units through contrast reversal. In E. Grandjean & E. Vigliani (Eds.), *Ergonomic aspects of visual display terminals* (pp. 137–142). London: Taylor & Francis.

Baumann, K., & Lanz, H. (1998). *Mensch-Maschine-Schnittstellen elektronischer Geräte. Leitfaden für Design und Schaltungstechnik* [Human-machine-interface for electronic appliances: Guideline for design and circuitry]. Berlin, Germany: Springer.

Bay, S., & Ziefle, M. (2004). Effects of menu foresight on information access in small screen devices. *48th annual meeting of the Human Factors and Ergonomic Society* (pp. 1841–1845). Santa Monica, CA: Human Factors Society.

Bergqvist, U., Wolgast, E., Nilsson, B., & Voss, M. (1995). The influence of VDT work on musculoskeletal disorders. *Ergonomics, 38*(4), 754–762.

Best, P., Littleton, M., Gramopadhye, A., & Tyrell, R. (1996). Relations between individual differences in oculomotor resting states and visual inspecting performance. *Ergonomics, 39*, 35–40.

Biocca, F. A., & Rolland, J. P. (1998). Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence, 7*(3), 262–277.

Blankenbach, K. (1999). Multimedia-Displays 2 Von der Physik zur Technik [Multimedia displays 2 from physics to technology]. *Physikalische Blätter, 55*(5), 33–38.

Blundel, B. G., & Schwarz, A. J. (2000). *Volumetric three-dimensional display systems*. New York: John Wiley & Sons.

Boff, K. R., & Lincoln, J. E. (1988). *Engineering data compendium—human perception and performance*. AAMRL, WPAFB, Ohio, Volumes 1–3. New York: John Wiley and Sons.

Boschmann, M., & Roufs, J. (1994). Reading and screen flicker. *Nature*, *372*, 137–142.

Bosman, D. (1989). *Display engineering: Conditioning, technologies, applications*. Amsterdam: Elesevier Science.

Brodie, J., Chattratichart, J., Perry, M., & Scane, R. (2003). How age can inform the future design of the mobile phone experience. In C. Stephanidis (Ed.), *Universal Access in HCI: Inclusive design in the information society* (pp. 822–826). Mahwah, NJ: LEA.

Budin J.-P. (2003). Emissive displays: the relative merits of ACTFEL. In L. MacDonald & A. Lowe (Eds.), *Display systems–design and applications* (2nd ed., pp. 191–219). New York: John Wiley & Sons.

Çakir, A., Hart, D. J., & Stewart, T. F. M. (1979). *The VDT manual–ergonomics, workplace design, health and safety, task organization*. Darmstadt, Germany: IFRA–Inca-Fiej Research Association.

Candry, P. (2003). Projection Systems. In L. MacDonald & A. Lowe (Eds.), *Display systems—design and applications* (2nd ed., pp. 237–256). New York: John Wiley & Sons.

Castellano, J. A. (1992). *Handbook of display technology*. San Diego, CA: Academic.

Dillon, A. (1992). Reading from paper versus screens: a critical review of the empirical literature. *Ergonomics, 35*(10), 1297–1326.

Duncan, J., & Humphreys, G. (1989). Visual search and stimulus similarity. *Psychological Review, 96*(3), 433–458.

E Ink Corporation (2001). *What is Electronic Ink?* Retrieved September 30, 2005, from http://www.eink.com/technology/index.htm

Feiner, S. K. (2002). Augmented reality–A new way of seeing. *Scientific American, 286*(4), 48–55.

Fruehauf, N., Aye, T., Yua, K., Zou, Y., & Savant, G. (2000). Liquid crystal digital scanner-based HMD. *Proceedings of SPIE Helmet- and Head-Mounted Displays V*, *4021*, 2–10, Orlando, FL, USA.

Gibbs, W. W. (1998). The reinvention of paper. *Scientific America Online*. Retrieved September 30, 2005, from http://www.sciam.com/1998/0998issue/0998techbus1.html

Goldstein, M., Öqvist, G., Bayat, M., Ljungstrand, P., & Björk, S. (2001). Enhancing the reading experience: Using adaptive and sonified RSVP for reading on small displays. *Proceedings of the Mobile HCI, 2001* (pp. 1–9). Berlin, Germany: Springer.

Gould, J., Alfaro, L., Barnes, V., Finn, R., Grischkowsky, N., & Minuto, A. (1987). Reading is slower from CRT displays than from paper. *Human Factors, 29*(3), 269–299.

Gröger, T., Ziefle, M., & Sommer, D. (2003). Anisotropic characteristics of LCD TFTs and their impact on visual performance. In D. Harris, V. Duffy, M. Smith, & C. Stephanidis (Eds.), *Human-centred computing: Cognitive, social and ergonomic aspects* (pp. 33–37). Mahwah, NJ: LEA.

Hanson, A. (2003). Measurements and standardization in the colorimetry of CRT displays. In L. MacDonald & A. Lowe (Eds.), *Display systems—Design and applications* (2nd ed., pp. 309–327). New York: John Wiley & Sons.

Helander, M., Little, S., & Drury, C. (2000). Adaptation and sensitivity to postural changes in sitting. *Human Factors, 42*(4), 617–629.

Heppner, F., Anderson, J., Farstrup, A., & Weidenman, N. (1985). Reading performance on standardized test is better from print than from computer display. *Journal of Reading, 28*, 321–325.

Hollands, J., Cassidy, H., McFadden, S., & Boothby, R. (2001). LCD versus CRT Displays: Visual search for colored symbols. *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting* (pp. 1353–1355). Santa Monica, CA: Human Factors Society.

Hollands, J., Parker, H., McFadden, S., & Boothby, R. (2002). LCD versus CRT Displays: A Comparison of visual search performance for colored symbols. *Human Factors, 44*(2), 210–221.

Holmes, R. (2003). Head-mounted display technology in virtual reality systems. In L. MacDonald & A. Lowe (Eds.), *Display systems—Design and applications*. (2nd., pp. 61–82). New York: John Wiley & Sons.

Holzel, T. (1999). Are head-mounted displays going anywhere? *Information Display, 15*(10), 16–18.

Howarth, P., & Istance, H. (1985). The association between visual discomfort and the use of visual display units. *Behaviour & Information Technology, 4*, 131–149.

Hung, G., Ciuffreda, K., & Semmlow J. (1986). Static vergence and accomodation: population norms and orthopic effects. *Documenta Ophthalmologica, 62*, 165–179.

ILT (2001). *Fraunhofer Institut für Lasertechnik (ILT)*. Retrieved October 15, 2005, from http://www.ilt.fhg.de

Iwasaki, T., & Kurimoto, S. (1988). Eye-strain and changes in accommodation of the eye and in visual evoked potential following quantified visual load. *Ergonomics, 31*(12), 1743–1751.

Jaschinski, W., Bonacker, M., & Alshuth, E. (1996). Accommodation, convergence, pupil and eye blinks at a CRT-display flickering near fusion limit. *Ergonomics, 19*, 152–164.

Jaschinski, W., Heuer, H., & Kylian, H. (1998). Preferred position of visual displays relative to the eyes: a field study of visual strain and individual differences. *Ergonomics, 41*, 1034–1049.

Kaufman, L. (1974). *Sight and mind 2 An introduction to visual perception*. London: Oxford University.

Kennedy, A., & Murray, W. (1993). 'Flicker' on the VDU screens. *Nature*, *365*, 213.

King, C. N. (1996). Electroluminescent displays, SID 96 Seminar Lecture Notes, M-9, 1–36.

Kokoschka, S., & Haubner, P. (1986). Luminance ratios at visual display workstations and visual performance. *Lighting Research & Technology, 17*(3), 138–144.

Kothiyal, K., & Tettey, S. (2001). Anthropometry for design for the elderly. *International Journal of Occupational Safety and Ergonomics, 7*(1), 15–34.

Küller, R., & Laike, T. (1998). The impact of flicker from fluorescent lighting on well-being performance and physiological arousal. *Ergonomics, 4* (4), 433–447.

Lieberman, D. (1999). Computer display clips onto eyeglasses. *Technology News*. Retrieved October 15, 2005, from http://www.techweb.com/wire/story/TWB19990422S0003

Lindner, H., & Kropf, S. (1993). Asthenopic complaints associated with fluorescent lamp illumination (FLI). *Lighting Research and Technology, 25*(2), 59–69.

Luczak, H., & Oehme, O. (2002). Visual Displays 2 Developments of the past, the present and the future. In H. Luczak, A. Çakir, & G. Çakir (Eds.). *Proceedings of the 6th International Scientific Conference on Work with Display Units* (pp. 2–5). Berlin, Germany: Ergonomic Institute.

Luczak, H., Park, M., Balazs, B., Wiedenmaier, S., & Schmidt, L. (2003). Task performance with a wearable augmented reality interface for welding. In D. Harris, V. Duffy, M. Smith, & C. Stephanidis (Eds). *Human-Computer Interaction. Cognitive, Social and Ergonomic Aspects* (pp. 98–102). Mahwah, NJ: LEA.

Lüder, E. (2003). Active matrix addressing of LCDs: merits and shortcomings. In L. MacDonald & A. Lowe (Eds.), Display systems—Design and applications (2nd ed., pp. 157–171). New York: John Wiley & Sons.

Macadam, D. L. (1982) Caliometry. *American Institute of Physics handbook* (pp. 6-182–6-197). New York: McGraw-Hill.

Matschulat, H. (1999). *Lexikon der Monitor-Technologie* [Lexicon of display technology]. Aachen, Germany: Elektor-Verlag

Menozzi, M., Lang, F., Näpflin, U., Zeller, C., & Krueger, H. (2001). CRT versus LCD: Effects of refresh rate, display technology and background luminance in visual performance. *Displays, 22*(3), 79–85.

Menozzi, M., Näpflin, U., & Krueger, H. (1999). CRT versus LCD: A pilot study on visual performance and suitability of two display technologies for use in office work. *Displays, 20*, 3–10.

Microvision (2000). *Homepage Microvision Inc*. Retrieved September 15, 2005, from http://www.mvis.com/

Miyao, M., Hacisalihzade, S., Allen, J., & Stark, L. (1989). Effects of VDT resolution on visual fatigue and readability: an eye movement approach. *Ergonomics, 32*(6), 603–614.

Nelson, T. J., & Wullert J. R. II (1997). *Electronic information display technologies* (Series on information display, Vol.3). River Edge, NJ: World Scientific.

Oehme, O., Schmidt, L., & Luczak, H. (2003). Comparison between the strain indicator HRV of a head-based virtual retinal display and LC head mounted displays for augmented reality. *International Journal of Occupational Safety and Ergonomics, 9*(4), 411–422.

Oehme, O., Wiedenmaier, S., Schmidt, L., & Luczak, H. (2001). Empirical studies on an augmented reality user interface for a head based virtual retinal display. In M. Smith, G. Salvendy, D. Harris, & R. Koubek (Eds.), *Usability evaluation and interface design: Cognitive engineering, intelligent agents and virtual reality* (pp. 1026–1030). Mahwah, NJ: LEA.

Oetjen, S., & Ziefle, M. (2004). Effects of anisotropy on visual performance regarding different font sizes. In H. Khalid, M. Helander, & A. Yeo (Eds.), *Work with computing systems* (pp. 442–447). Kuala Lumpur, Malaysia: Damai Sciences.

Oetjen, S., Ziefle, M., & Gröger, T. (2005). Work with visually suboptimal displays—in what ways is the visual performance influenced when CRT and TFT displays are compared? In *Proceedings of the HCI International 2005: Vol. 4: Theories, Models and Processes in Human Computer Interaction*. St. Louis, MO: Mira Digital Publisher.

Oetjen, S., & Ziefle, M. (in press). The effects of LCD anisotropy on the visual performance of users of different ages. *Human Factors, 49*(4).

Omori, M., Watanabe, T., Takai, J., Takada, H., & Miyao, M. (2002). Visibility and characteristics of the mobile phones for elderly people. *Behaviour & Information Technology, 21*(5), 313–316. St. Louis (MO).

Ono, Y. A. (1993) Electroluminescent displays, Seminar Lecture Notes, F-1/1-30, Society for Information Display, Santa Ana: CA.

Owens, D., & Wolf- Kelly, K. (1987). Near work, visual fatigue, and variations of oculomotor tonus. *Investigative Ophthalmology and Visual Science, 28*(4), 743–749.

Park, M., Schmidt, L., & Luczak, H. (2005). Changes in hand-eye-coordination with different levels of camera displacement from natural eye position. In *10th International Conference on Human Aspects of Advanced Manufacturing: Agility and Hybrid Automation 2 HAAMAHA 2005*, San Diego, CA, USA, 191–199.

Pausch, R., Dwivedi, P., & Long, J. A. C. (1991). A practical, low-cost stereo head-mounted display. *Proceedings of SPIE. Stereoscopic Displays and Applications II*, *1457*, 198–208.

Pfendler, C., Widdel, H., & Schlick, C. (2005). *Bewertung eines Head-Mounted und eines Hand-Held Displays bei einer Zielerkennungsaufgabe* [Task related evaluation of head-mounted and hand-held displays]. *Zeitschrift für Arbeitswissenschaft, 59*(1), 13–21.

Pfendler, C., & Schlick, C. (in press). A comparative study of mobile map displays in a geographic orientation task. *Behaviour and Information Tecchnology, 26*(2).

Philips Research Press Information (2005). *Philips demonstrates photonic textiles that turn fabric into intelligent displays*. Retrieved October 15, 2005, from http://www.research.philips.com/newscenter/archive/2005/050902-phottext.html

Pioneer (2001). *Why choose Plasma*. Retrieved September 15, 2005, from http://www.pioneerelectronics.com/Pioneer/CDA/Common/ArticleDetails/0,1484,1547,00.html

Plainis, S., & Murray, I. (2000). Neurophysiological interpretation of human visual reaction times: effect of contrast, spatial frequency and luminance. *Neuropsychologia, 38*, 1555–1564.

Precht, M., Meier, N., & Kleinlein, J. (1997). *EDV-Grundwissen: Eine Einführung in Theorie und Praxis der modernen EDV* [Computer basics: An introduction in theory and application of modern computing]. Bonn, Germany: Addison-Wesley-Longman.

Rahman, T., & Muter, P. (1999). Designing an interface to optimize reading with small display windows. *Human Factors, 41*(1), 106–117.

Sanders, M. S., & McCormick, E. J. (1993). Human Factors in Engineering and Design (7th ed.). New York: McGraw-Hill.

Schadt, M. (1996). *Optisch strukturierte Flüssigkeitskristall-Anzeigen mit großem Blickwinkelbereich* [Optically structured liquid crystal displays with wide viewing angles]. *Physikalische Blätter, 52*(7/8), 695–698.

Schenkmann, B., Fukunda, T., & Persson, B. (1999). Glare from monitors measured with subjective scales and eye movements. *Displays, 20*, 11–21.

Schlick, C., Daude, R., Luczak, H., Weck, M., & Springer, J. (1997): Head-mounted display for supervisory control in autonomous production cells. *Displays, 17*(3/4), 199–206.

Schlick, C., Reuth, R., & Luczak, H. (2000). Virtual Reality User Interface for Autonomous Production. In: L. M. Camarinha-Matos, H. Afsarmanesh, & H. Erbe (Eds.), *Advances in Networked Enterprises* (pp. 279–286). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Schlick, C., Winkelholz, C., Motz, F., & Luczak, H. (2006). Self-Generated Complexity and Human-Machine Interaction. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 36*(1), 220–232.

Sheedy, E. J. (2005). Office Lighting for Computer Use. In Anshel, J. (Ed.), *Visual ergonomics handbook* (pp. 37–51). Boca Raton, FL: CRC.

Sheedy, J., Subbaram, M., & Hayes, J. (2003). Filters on computer displays-effects on legibility, performance and comfort. *Behaviour and Information Technology, 22*(6), 427–433.

Sommerich, C., Joines, S., & Psihoios, J. (2001). Effects of computer monitor viewing angle and related factors on strain, performance, and preference outcomes. *Human Factors, 43*(1), 39–55.

Stanney, K. M., & Zyda, M. (2002) Virtual environments in the 21st century. In Stanney, K. M. *Handbook of virtual environments* (pp. 1–14). Mahwah, NJ: LEA.

Stone, P., Clarke, A., & Slater, A. (1980). The effect of task contrast on visual performance and visual fatigue at a constant illuminance. *Lighting Research and Technology, 12*, 144–159.

Theis, D. (1999). Display Technologie [Display technology]. In C. Müller-Schloer & B. Schallenberger (Eds.), *Vom Arbeitsplatzrechner zum ubiquitären Computer* [From the desktop PC to the ubiquitous computer]. (pp. 205–238), Berlin, Germany: VDE.

Urey, H., Wine, D. W., & Lewis J. R. (1999). Scanner design and resolution tradeoffs for miniature scanning displays. In B. Gnade & E. F. Kelley (Eds.), *Proceedings SPIE, Flat Panel Display Technology and Display Metrology 3636,* San Jose, CA, USA, 60–68.

van Schaik, P., & Ling, J. (2001). The effects of frame layout and differential background contrast on visual search performance in web pages. *Interacting with Computers, 13*, 513–525.

von Waldkirch, M. (2005). *Retinal projection displays for accommodation-insensitive viewing.* Unpublished doctoral thesis, Zurich, Swiss Federal Institute of Technology.

Weiss, S. (2002). *Handheld Usability*. New York: John Wiley.

Wilkins, A., Nimmo-Smith, I., Tait, A., McManus, C., Della Sala, S., Tilley, A., et al. (1984). A neurological basis for visual discomfort. *Brain, 107*, 989–1017.

Wolf, E., & Schraffa, A. (1964). Relationship between critical flicker frequency and age in flicker perimetry. *Archives of Opbthalmology, 72*, 832–843.

Woodson, W. E. (1987). *Human factor reference guide for electronics and computer professionals*. New York: McGraw-Hill.

Yeh, Y.-Y., & Wickens, C. (1984). Why do performance and subjective workload measures dissociate? *Proceedings of the 28th Annual meeting of the Human Factors Society* (pp. 504–508). Santa Monica, CA: Human Factors and Ergonomics Society.

Ziefle, M., & Bay, S. (2005). How older adults meet cognitive complexity: Aging effects on the usability of different cellular phones. *Behavior & Information Technology, 24*(5), 375–389.

Ziefle, M., & Bay, S. (2006). How to overcome disorientation in mobile phone menus: A comparison of two different types of navigation aids. *Human Computer Interaction, 21*(4), 393–433.

Ziefle, M. (1998). Effects of display resolution on visual performance. *Human Factors, 40*(4), 554–568.

Ziefle, M. (2001a). CRT screens or TFT displays? A detailed analysis of TFT screens for reading efficiency. In M. Smith, G. Salvendy, D. Harris, & R. Koubek (Eds.), *Usability evaluation and interface design* (pp. 549–553). Mahwah, NJ: LEA.

Ziefle, M. (2001b). Aging, visual performance and eyestrain in different screen technologies. *Proceedings of the Human Factors and Ergonomics Society 45th annual meeting* (pp. 262–266). Santa Monica, CA: Human Factors Society.

Ziefle, M. (2002). *Lesen am Bildschirm* [Reading from screens]. Münster, Germany: Waxmann.

Ziefle, M. (2003a). Users with body heights above and below the average: How adequate is the standard VDU setting with respect to visual performance and muscular load? In H. Luczak & K. J. Zink (Eds.), *Human Factors in Organizational Design and Management* (pp. 489–494). Santa Monica, CA: IEA Press.

Ziefle, M. (2003b). Sitting posture, postural discomfort, and visual performance: A critical view on the independence of cognitive and anthropometric factors in the VDU workplace. *International Journal of Occupational Safety and Ergonomics*, *9*(4), 495–506.

Ziefle, M., Gröger, T., & Sommer, D. (2003). Visual costs of the inhomogeneity of contrast and luminance by viewing LCD-TFT screens off-axis. *International Journal of Occupational Safety and Ergonomics, 9*(4), 507–517.

Ziefle, M., Oehme, O., & Luczak, H. (2005). Information presentation and visual performance in head-mounted displays with augmented reality. *Zeitschrift für Arbeitswissenschaft, 59*(3–4), 331–344.

12

HAPTIC INTERFACE

Hiroo Iwata
University of Tsukuba

Mechanism of Haptics and Methods for Haptic Feedback **230**
Somatic Sensation 230
Proprioception and Force Display 230
(1) Exoskeleton-Type of Force Display 230
(2) Tool-Handling Type of Force Display 230
(3) Object-Oriented Type of Force Display 231
(4) Passive Prop 231
Proprioception and Full-Body Haptics 231
Sliding Device 231
Treadmill 232
Foot Pads 232
Pedaling Device 232
Gesture Recognition of Walking 233
Skin Sensation and Tactile Display 233
Technologies in Finger/Hand Haptics **233**
Exoskeleton 233
Tool-Handling-Type of Haptic Interface 233
Technologies in Finger/Hand Output: Object-Oriented-Type Haptic Interface **234**
Basic Idea of FEELEX 234
Design Specification and Implementation of Prototypes **235**
FEELEX 1 235
FEELEX 2 236
Characteristics of the FEELEX 237
Volmetric Object-Oriented-Type of Haptic Interface **237**
Basic Design of Volflex 237
Virtual Clay Volflex 238
Technologies in Full-Body Haptics **238**
Treadmill-Based Locomotion Interface 238
Basic Design of the Torus Treadmill 238
Mechanism and Performance 238
Control Algorithm of the Torus Treadmill 239
Foot-Pad-Based Locomotion Interface 240
Methods of Presentation of Uneven Surface 240
Basic Design of the GaitMaster 240
Control Algorithm of the GaitMaster 241
Prototype GaitMaster 241
Robot-Tile-Based Locomotion Interface 242
CirculaFloor Project 242
Method of Creating Infinite Surface 242
Application Area for Haptic Interface **243**
Application Area for Finger/Hand Haptics 243
Medicine 243
3D-shape modeling 243
HUI (Haptic User Interface) 243
Art 243
Application Areas for Locootion Interfaces 243
Conclusion and Future Prospects **244**
Safety Issues 244
Psychology in Haptics 244
References **244**

It is well known that sense of touch is inevitable for understanding the real world. The use of force feedback to enhance computer-human interaction (CHI) has often been discussed. A Haptic interface is a feedback device that generates sensation to the skin and muscles, including a sense of touch, weight, and rigidity. Compared to ordinary visual and auditory sensations, a haptic is difficult to synthesize. Visual and auditory sensations are gathered by specialized organs, the eyes and ears. On the other hand, a sensation of force can occur in any part of the human body and is therefore inseparable from actual physical contact. These characteristics lead to many difficulties when developing a haptic interface. Visual and auditory media are widely used in everyday life, although little application of haptic interface is used for information media.

In the field of virtual reality, haptic interface is one of the major research areas. The last decade has seen significant advance in development of haptic interface. High-performance haptic devices have been developed and some of them are commercially available. This chapter presents current methods and issues in haptic interface.

The next section describes the mechanism of haptic sensation and overall view of feedback technologies. This section is followed by three sections that introduce examples of haptic-interface technologies developed by the author. The last section presents application areas and future prospects of haptic interface.

MECHANISM OF HAPTICS AND METHODS FOR HAPTIC FEEDBACK

Somatic Sensation

Haptic interface presents synthetic stimulation to somatic sensation. Somatic sensation is composed of proprioception and skin sensation. Proprioception is complemented by mechanoreceptors of skeletal articulations and muscles. There are three types of joint position receptors: (a) free nerve ending as well as (b) Ruffini and (c) Pacinian corpuscles. Ruffini corpuscle detects static force. On the other hand, Pacinian corpuscle has a function to measure acceleration of the joint angle. Position and motion of the human body is perceived by these receptors. Force sensation is derived from mechanoreceptors of muscles; muscle spindles and Goldi tendons. These receptors detect contact forces applied by an obstacle in the environment.

Skin sensation is derived from mechanoreceptors and thermorecepters of skin. Sense of touch is evoked by those receptors. Mechanoreceptors of skin are classified into four types: (a) Merkel disks, (b) Ruffini capsules, (c) Meissner corpuscles, and (d) Pacinian corpuscles. These receptors detect edges of objects, skin stretching, velocity, and vibration, respectively.

Proprioception and Force Display

Force display is a mechanical device that generates a reaction force from virtual objects. Recently, much research has been conducted on haptic interfaces, although the technology is still in a state of trial and error. There are several approaches to implementing haptic interfaces:

Exoskeleton-Type of Force Display

An *exoskeleton* is a set of actuators attached to a hand or a body. In the field of robotics research, exoskeletons have often been used as master manipulators for teleoperations. However, most master manipulators entail a large amount of hardware and therefore have a high cost, which restricts their application areas. Compact hardware is needed to use them in HCI. The first example of a compact exoskeleton suitable for desktop use was published in 1990 (Iwata, 1990). The device applies force to the fingertips as well as the palm. Figure 12.1 shows an overall view of the system.

Lightweight and portable exoskeletons have also been developed. Burdea used small pneumatic cylinders to apply the force to the fingertips (Burdea, Zhuang, Roskos, Silver, & Langlana, 1992). Cyber Grasp (Fig. 12.2) is a commercially available exoskeleton, in which cables are used to transmit force [http://www.vti.com].

Tool-Handling Type of Force Display

The tool-handling type of force display is the easiest way to realize force feedback. The configuration of this type is similar to that of a joystick. Unlike the exoskeleton, the tool-handling-type force display is free from the need to be fitted to the user's hand. It cannot generate a force between the fingers, but has practical advantages.



FIGURE 12.1. Desktop force display.



FIGURE 12.2. Cyber Grasp.



FIGURE 12.3. PHANToM.

A typical example of this category is the pen-based force display (Iwata, 1993). A pen-shaped grip is supported by two 3DOF pantographs that enable a 6DOF force/torque feedback. Another example of this type is the Haptic Master, which was demonstrated at the edge venue of SIGGRAPH'94. The device has a ball-shaped grip to which 6 DOF force/torque is fed back (Iwata, 1994). This device employs a parallel mechanism in which a top triangular platform and a base triangular platform are connected by three sets of pantographs. This compact hardware has the ability to carry a large payload.

Massie and Salisbury developed the PHANToM, which has a 3 DOF pantograph (Massie & Salisbury, 1994). A thimble with a gimbal is connected to the end of the pantograph, which can then apply a 3 DOF force to the fingertips. The PHANToM became one of the most popular commercially available haptic interfaces (Fig. 12.3).

Object-Oriented Type of Force Display

The object-oriented type of force display is a radical idea for the design of a haptic interface. The device moves or deforms to simulate the shapes of virtual objects. A user can make physical contact the surface of the virtual object.

An example of this type can be found in Tachi and colleagues' work (Tachi et al., 1994). Their device consists of a shape-approximation prop mounted on a manipulator. The position of the fingertip is measured and the prop moves to provide a contact point for the virtual object. McNeely proposed an idea named "Robotic Graphics" (McNeely, 1993), which is similar to Tachi and colleagues' method. Hirose developed a surface display that creates a contact surface using a 4×4 linear actuator array (Hirota & Hirose, 1996). The device simulates an edge or a vertex of a virtual object.

Passive Prop

A passive input device equipped with force sensors is a different approach to the haptic interface. Murakami and Nakajima used a flexible prop to manipulate a three dimensional (3D) virtual object (Murakami & Nakajima, 1994). The force applied by the user is measured and the deformation of the virtual object is determined based on the applied force. Sinclair developed a force-sensor array to measure pressure distribution (Sinclair, 1997). These passive devices allow users to interact using their bare fingers. However, these devices have no actuators, so they cannot represent the shape of virtual objects.

Proprioception and Full-Body Haptics

One of the new frontiers of haptic interface is full-body haptics that includes foot haptics. Force applied to a whole body plays a very important role in locomotion. The most intuitive way to move about the real world is walking on foot. *Locomotion interface* is a device that provides a sense of walking while the walker's body is localized in the real world. There are several approaches to realizing locomotion interface:

Sliding Device

A project named "Virtual Perambulator" developed locomotion interface using a specialized sliding device (Iwata & Fujii, 1996). The primary objective of the first stage was to allow the walker's feet to change direction. Controlling steering bars or joysticks are not as intuitive as in locomotion. The first prototype of the Virtual Perambulator was developed in 1989 (Iwata & Matsuda, 1990). Figure 12.4 shows overall view of the apparatus.



FIGURE 12.4. The first prototype of Virtual Perambulator.

A user wore a parachute-like harness and omni-directional roller skates. The trunk of the walker was fixed to the framework of the system by the harness. An omni-directional sliding device is used for changing direction by foot. Specialized roller skates equipped with four casters were developed, which enabled two-dimensional motion. The walker could freely move his or her feet in any direction. Motion of the feet was measured by ultrasonic range detector. From the result of this measurement, an image of the virtual space was displayed in the head-mounted display corresponding with the motion of the walker. The direction of locomotion in virtual space was determined according to the direction of the walker's step.

Treadmill

A simple device for virtual walking is a treadmill, a device ordinarily used for physical fitness. An application of this device used as a virtual building simulator was developed at UNC (Brooks, 1986). This treadmill has a steering bar similar to that of a bicycle. A treadmill equipped with a series of linear actuators underneath the belt was developed at ATR (Noma, Sugihara, & Miyasato, 2000). The device was named GSS, and it simulates the slope of virtual terrain. The Treadport developed at the University of Utah is a treadmill combined with a large manipulator connected to a walker (Christensen, 1998). The manipulator provides gravitational force while the walker is passing a slope. Figure 12.5 shows overall view of the TreadProt.

The omni-directional treadmill employs two perpendicular treadmills, one inside of the other. Each belt is made from approximately 3,400 separate rollers, which are woven together into a mechanical fabric. Motion of the lower belt is transmitted to the walker with rollers. This mechanism enables omni-directional walking (Darken, Cockayne, & Carmein, 1997).

Foot Pads

A foot pad applied to each foot is an alternative implementation of locomotion interface. Two large manipulators driven by hydraulic actuators were developed at University of Utah and applied to a locomotion interface. These manipulators are attached to a walker's feet. The device is named BiPort [http://www.sarcos.com]. The manipulators can present viscosity of virtual ground. A similar device has been developed at the Cybernet Systems Corporation, which uses two 3 DOF motion platforms for the feet (Poston et al., 1997). These devices, however, have not been evaluated or applied to VE.

Pedaling Device

In the battlefield simulator of NPSNET project, a unicycle-like pedaling device is used for locomotion in the virtual battlefield (Plat et al., 1994). A player changes direction by twisting his or her waist.

The OSIRIS, a simulator of night-vision battle, utilizes a stair-stepper device (Lorenzo et al., 1997). A player changes direction using a joystick or by twisting his or her waist.



FIGURE 12.5. Treadport.

Gesture Recognition of Walking

Slaters et al. proposed locomotion in virtual environments by "walking in place." They recognized the gesture of walking using a position sensor and a neural network (Slater et al., 1994).

Skin Sensation and Tactile Display

The tactile display that stimulates skin sensation is a well-known technology. It has been applied to communication aids for blind persons as well as a master system of tele-operation. A sense of vibration is relatively easy to produce, and a good deal of work has been done using vibration displays (Kontarinis & Howe, 1995; Minsky & Lederman, 1997). The micropin array is also used for tactile displays. Such a device has enabled the provision of a teltaction and communication aid for blind persons (Moy et al., 2000; Kawai & Tomita, 2000). It has the ability to convey texture or 2D-geometry (Burdea, 1996).

The micropin array looks similar to the object-oriented-type force display, but it can only create the sensation of skin. The stroke distance of each pin is short, so the user cannot feel the 3D-shape of a virtual object directly. The major role of tactile display is to convey the sense of fine texture of an object's surface. The latest research activities of tactile display focus on selective stimulation of mechanoreceptors of the skin. As mentioned at the beginning of this section, there are four types of mechanoreceptors in the skin: (a) Merkel disks, (b) Ruffini capsules, (c) Meissner corpuscles, and (d) Pacinian corpuscles. Stimulating these receptors selectively, various tactile sensations such as roughness or slip can be presented. Microair jet (Asanura et al.,1999) and microelectrode array (Kajimoto et al., 1999) are used for selective stimulation.

TECHNOLOGIES IN FINGER/HAND HAPTICS

Exoskeleton

An exoskeleton is one of the typical forms of haptic interface. Figure 12.6 shows detailed view of an exoskeleton (Iwata, 1990).

Force sensation contains six-dimensional information: three-dimensional forces and three-dimensional torque. The core element of the force display is a six-degree-of-freedom parallel manipulator. The typical design feature of parallel manipulators is an octahedron called "Stewart platform." In this mechanism, a top triangular platform and a base triangular platform are connected by six length-controllable cylinders. This compact hardware has the ability to carry a large payload. The structure, however, has some practical disadvantages in its small-working volume and its lack of backdrivability (reduction of friction) of the mechanism. In our system, three sets of parallelogram linkages (pantograph) are employed instead of linear actuators. Each pantograph is driven by two DC motors. Each motor is powered by a PWM (pulse width modulation) amplifier. The top end of the pantograph is connected to the vertex of the top platform with a spherical joint. This mechanical configuration has the same advantages as an octahedron mechanism. The pantograph

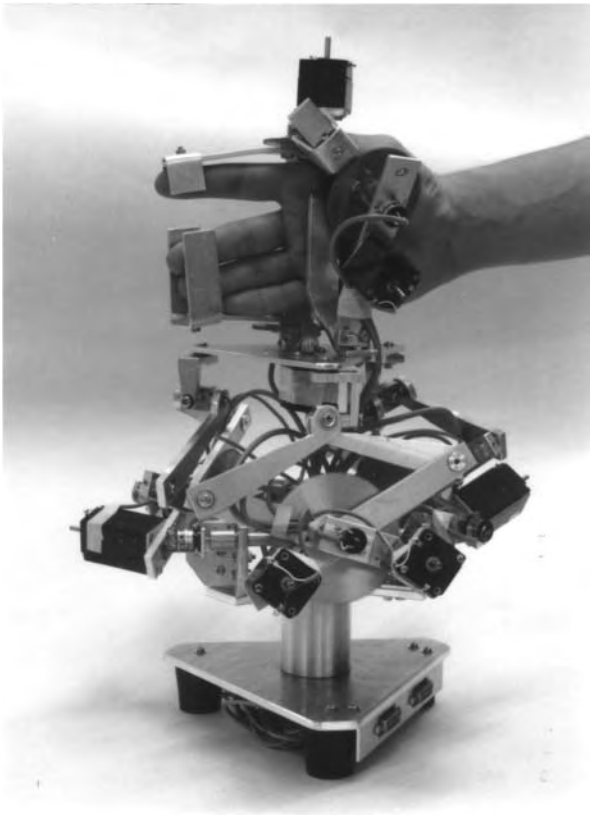


FIGURE 12.6. Desktop force display.

mechanism improves the working volume and backdrivability of the parallel manipulator. The inertia of motion parts of the manipulator are so small that compensation is not needed.

The working space of the center of the top platform is a spherical volume whose diameter is approximately 30 cm. Each joint angle of the manipulator is measured by potentiometers. Linearity of the potentiometers is 1%. The maximum payload of the manipulator is 2.3 kg, which is more than a typical hand.

The top platform of the parallel manipulator is fixed to the palm of the operator with a U-shaped attachment, which enables the operator to move the hand and fingers independently. Three actuators are set coaxially with the first joint of the thumb, forefinger, and middle finger of the operator. The last three fingers work together. DC servo motors are employed for each actuator.

Tool-Handling-Type of Haptic Interface

Exoskeletons are troublesome for users to put on or take off. This disadvantage obstructs practical use of force displays. A tool-handling-type of haptic interface implements force display without a glove-like device. A pen-based force display is proposed as an alternative device (Iwata, 1993). A six-degree-of-freedom force reflective master manipulator, which has pen-shaped grip, was developed. Users are familiar with pens because most human intellectual works are done in pen. In this aspect,

the pen-based force display is easily applied to the design of 3D shapes.

The human hand has the ability of six-degree-of-freedom motion in 3D space. If a six-degree-of-freedom master manipulator is built using serial joints, each joint must support the weight of upper joints. Because of this, manipulator hardware must be large; we use parallel mechanism in order to reduce size and weight of the manipulator. The pen-based force display employs two three-degree-of-freedom manipulators. Both ends of the pen are connected to these manipulators. Total degree-of-freedom of the force display is six. Three-degree-of-freedom force and three-degree-of-freedom torque are applied at the pen. An overall view of the force display is shown in Fig. 12.7. Each three-degree-of-freedom manipulator is composed of pantograph link. By this mechanism, the pen is free from the weight of the actuators.

Figure 12.8 shows a diagram of mechanical configuration of the force display. Joints MA1, MA2, MA3, MB1, MB2, and MB3 are equipped with DC motors and potentiometers. Other joints move passively. The position of joint A and B is measured by potentiometers. A three-dimensional force vector is applied at the joint A and B. The joint A determines the position of the pen point, and the joint B determines the orientation of the pen. Working space of the pen point is a part of a spherical volume whose diameter is 44 cm. The rotational angle around the axis of the pen is determined by the distance between joint A and B. A screw-motion mechanism converts rotational motion of the pen into transition of the distance between joint A and B.

Applied force and torque at the pen is generated by combination of forces at the point A and B. In case these force have the same direction, translational force is applied to the user's hand. If direction of the forces are reverse, torque around the yaw axis or the pitch axis is generated. If two forces are opposite, torque around the roll axis is generated by the screw-motion mechanism.

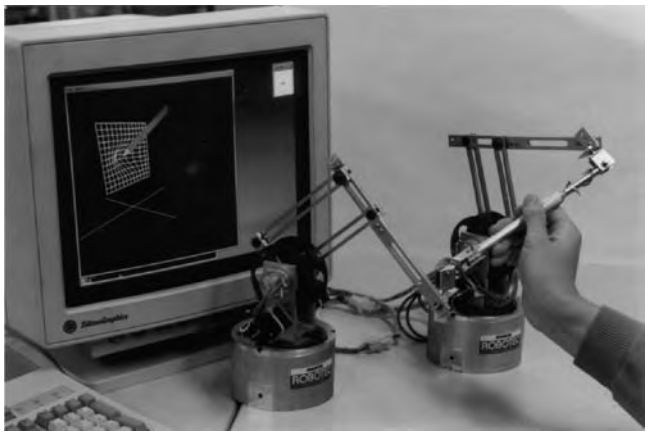


FIGURE 12.7. Pen-based force display.

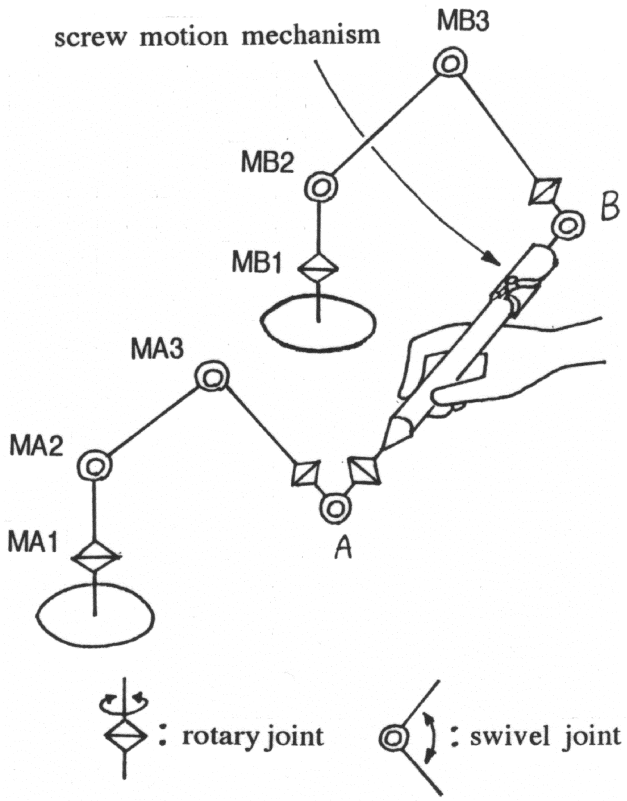


FIGURE 12.8. Mechanical configuration of pen-based force display.

TECHNOLOGIES IN FINGER/HAND OUTPUT: OBJECT-ORIENTED-TYPE HAPTIC INTERFACE

Basic Idea of FEELEX

The author demonstrated haptic interfaces to a number of people, and found that some of them were unable to fully experience virtual objects through the medium of synthesized haptic sensation. There seem to be two reasons for this phenomenon. First, these haptic interfaces only allow the users to touch the virtual object at a single point or at a group of points. These contact points are not spatially continuous, due to the hardware configuration of the haptic interfaces. The user feels a reaction force through a grip or thimble. Exoskeletons provide more contact points, but these are achieved using Velcro bands attached to a specific part of the user's fingers, which are not continuous. Therefore, these devices cannot recreate a natural interaction sensation when compared to manual manipulation in the real world.

The second reason why users fail to perceive the sensation is related to a combination of the visual and haptic displays. A visual image is usually combined with a haptic interface using a conventional CRT or projection screen. Thus, the user receives visual and haptic sensation through different displays, and therefore has to integrate the visual and haptic images in his or her brain. Some users, especially elderly people, have difficulty in this integration process.

Considering these problems, new interface devices have developed. The project is named "FEELEX." The word FEELEX is derived from a conjunction of "feel" and "flex." The major goals of this project are,

- to provide a spatially continuous surface that enables users to feel virtual objects using any part of the fingers or even the whole palm
- to provide visual and haptic sensations simultaneously using a single device that doesn't oblige the user to wear any extra apparatus

A new configuration of visual/haptic display was designed to achieve these goals. Figure 12.9 illustrates the basic concept of the FEELEX. The device is composed of a flexible screen, an array of actuators, and a projector. The flexible screen is deformed by the actuators in order to simulate the shape of virtual objects. An image of the virtual objects is projected onto the surface of the flexible screen. Deformation of the screen converts the 2D image from the projector into a solid image. This configuration enables users to touch the image directly using any part of their hand. The actuators are equipped with force sensors to measure the force applied by the user. The hardness of the virtual object is determined by the relationship between the measured force and its position on the screen. If the virtual object is soft, a large deformation is caused by a small, applied force.

DESIGN SPECIFICATION AND IMPLEMENTATION OF PROTOTYPES

FEELEX 1

The FEELEX 1, developed in 1997, was designed to enable double-handed interaction using the whole palms. Therefore, the optimum size of the screen was determined to be 24 cm x 24 cm. The screen is connected to a linear actuator array that deforms its shape. Each linear actuator is composed of a screw mechanism driven by a DC motor. The screw mechanism converts the rotation of an axis of the motor to the linear motion of a rod. The motor must generate both motion and a reaction force on the screen. The diameter of the smallest motor that can drive the screen is 4 cm. Therefore, a 6 x 6 linear-actuator array can be set under the screen. The deformable screen is made of a rubber plate and a white nylon cloth. The thickness of the rubber is 3 mm. Figure 12.10 shows an overall view of the device.

The screw mechanism of the linear actuator has a self-lock function that maintains its position while the motor power is off. Hard virtual wall is difficult to simulate using tool-handling-type force displays. Considerable motor power is required to generate the reaction force from the virtual wall, which often leads to uncomfortable vibrations. The screw mechanism is free from this problem. A soft wall can be represented by the computer-

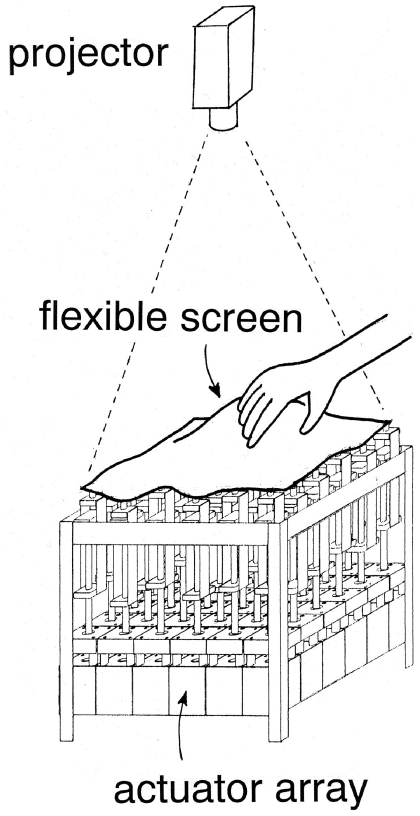


FIGURE 12.9. Basic design of FEELEX.



FIGURE 12.10. Overall view of the FEELEX 1.

controlled motion of the linear actuators based on the data from the force sensors. A force sensor is set at the top of each linear actuator. Two strain gauges are used as a force sensor. The strain gauge detects small displacements of the top end of the linear actuator caused by the force applied by the user. The position of the top end of the linear actuator is measured by an optical encoder connected to the axis of the DC motor. The maximum stroke of the linear actuator is 80 mm, and the maximum speed is 100 mm/s.

The system is controlled via a personal computer (PC). The DC motors are interfaced by a parallel I/O unit, and the force sensors are interfaced by an A/D converter unit. The force sensors provide interaction with the graphics. The position and strength of the force applied by the user are detected by a 6×6 sensor array. The graphics projected onto the flexible screen are changed according to the measured force.

FEELEX 2

The FEELEX 2 is designed to improve the resolution of the haptic surface. In order to determine the resolution of the linear actuators, we considered the situation where a medical doctor palpates a patient. After interviewing several medical doctors, it was discovered that they usually recognized a tumor using their index finger, middle finger, and third finger. The size of a tumor is perceived by comparing it to the width of their fingers, i.e. two-fingers large or three-fingers large. Thus, the distance between the axis of the linear actuators should be smaller than the width of a finger. Considering the condition just described, the distance is set to be 8 mm. This 8 mm resolution enables the user to hit at least one actuator when he or she touches any arbitrary position on the screen. The size of the screen is 50 mm $\times$ 50 mm, which allows the user to touch the surface using three fingers.

In order to realize 8 mm resolution, a piston-crank mechanism is employed for the linear actuator. The size of the motor is much larger than 8 mm, so the motor should be placed at a position offset from the rod. The piston-crank mechanism can easily achieve this offset position. Figure 12.11 illustrates the mechanical configuration of the linear actuator. A servo-motor from a radio-controlled car is selected as the actuator. The rotation of the axis of the servo-motor is converted to the linear motion of the rod by a crank-shaft and a linkage. The stroke of the rod is 18 mm, and the maximum speed is 250 mm/s. The maximum torque of the servo-motor is 3.2 Kg-cm, which applies a 1.1 Kgf force at the top of each rod. This force is sufficient for palpation using the fingers.

The flexible screen is supported by 23 rods, and the servo-motors are set remotely from the rods. Figure 12.12 shows an overall view of the FEELEX 2. The 23 separate sets of piston-crank mechanisms can be seen in the picture.

Figure 12.13 shows the top end of the rods. The photo was taken while the flexible screen is off. The diameter of each rod is 6 mm. A strain gauge cannot be put on top of the rod because of its small size. Thus, the electric current going to each servo-motor is measured to sense the force. The servo-motor generates a force to maintain the position of the crank shaft. When the user applies force to the rod, the electric current on the mo-

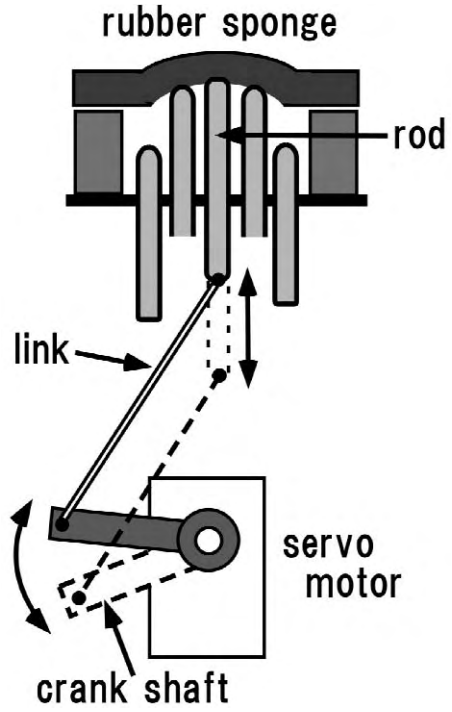


FIGURE 12.11. Piston-crank mechanism.

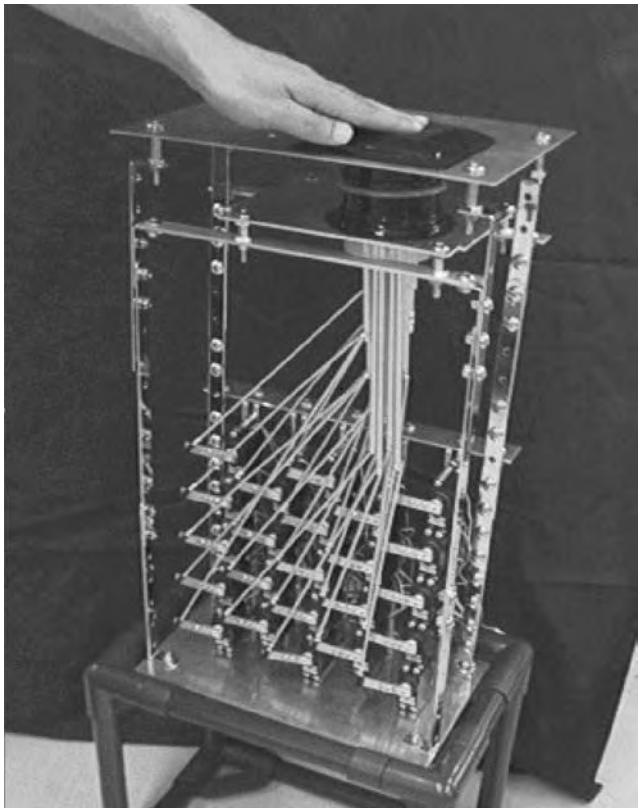


FIGURE 12.12. Overall view of the FEELEX 2.



FIGURE 12.13. Top end of the rods.

tor increases to balance the force. The relationship between the applied force and the electric current is measured. The applied force at the top of the rods is calculated using data from the electric current sensor. The resolution of the force sensing capability is 40 gf.

Characteristics of the FEELEX

The performance of existing haptic interfaces is usually represented by the dynamic range of force, impedance, inertia, friction, etc. However, these parameters are only crucial while the device is attached to the finger or hand. In the case of the tool-handling-type of haptic interface or the exoskeleton, the devices move with the hand even though the user doesn't touch the virtual objects. Therefore, inertia or friction degrades the usability and the dynamic range of force determines the quality of the virtual surface. On the other hand, the FEELEX is entirely separate from the user's hand, so its performance is determined by the resolution and speed of the actuators. The resolution of the actuator corresponds to the smoothness of the surface and the speed of the actuator determines the motion of the virtual object. The FEELEX 2 has improved resolution and motion speed compared to the FEELEX 1. Each actuator of the FEELEX 2 has a stroke rate of up to 7 Hz, which can simulate the motion of a very fast virtual object. The rod pushes the rubber sponge so that the user feels as if the object was pulsating. 7 Hz is much faster than the human pulse rate.

The major advantage of the FEELEX is that it allows natural interaction using only the bare hand. In SIGGRAPH'98, 1,992 subjects spontaneously enjoyed the haptic experience. One of the subject contents of the FEELEX 1 system, known as Anomalocaris, was selected as a long-term exhibition at the Ars Electronica Center (Linz, Austria). The exhibition has been popular among visitors, and especially children.

Another advantage of FEELEX is safety. The user of FEELEX doesn't wear any special equipment while the interaction is taking place. The exoskeleton and tool-handling-type force displays have control problems in their contact surface for the virtual objects. Vibration or unwanted forces can be generated back to the user, which is sometimes dangerous. The contact surface of the FEELEX is physically generated, so it is free from such control problems.

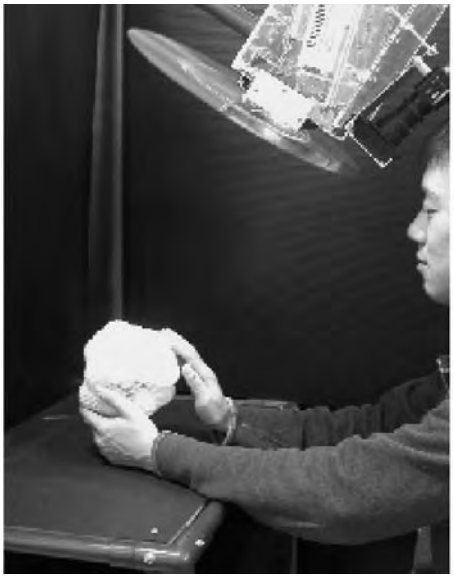


FIGURE 12.14. Overall view of Volflex.

The major disadvantage of the FEELEX is the degree of difficulty in its implementation. It requires a large number of actuators that have to be controlled simultaneously. The drive mechanism of the actuator must be robust enough for rough manipulation. Since the FEELEX provides a feeling of natural interaction, some of the users apply large forces. Our exhibit at the Ars Electronica Center suffered from overload of the actuators.

Another disadvantage of the FEELEX is its limitation in the shape of objects that can be displayed. The current prototypes cannot present a sharp edge on a virtual object. Furthermore, the linear actuator array can only simulate the front face of objects. Some of the participants of the Anomalocaris demonstration wanted to touch the rear of the creature, but an entirely new mechanism would be required to simulate the reverse side of the object.

VOLMETRIC OBJECT-ORIENTED-TYPE OF HAPTIC INTERFACE

Basic Design of Volflex

In order to present a side or the back of a virtual object, we designed a volumetric object-oriented haptic interface. Volflex is a new haptic interface that provides the user with a physical 3D surface for interaction. The device is composed of a group of air balloons. The balloons fulfill the interaction surface, and are arranged in a body-centered cubic lattice. A tube is connected to each balloon, and the volume of each balloon is controlled by an air cylinder. The tubes are connected to each other by springs. This mechanical flexibility enables an arbitrary shape of the in-



FIGURE 12.15. Examples of deformation.

teraction surface. Each air cylinder is equipped with a pressure sensor that detects force applied by the user. According to the pressure data, the device is programmed to perform like a clay. Unlike real clay, Volflex allows the user to "Un do" an operation.

A projector is set above the balloons. An image is projected onto the surface of the device, not on the user's hand. We developed a mechanical rotary shutter that separates the projector and camera. The camera captures the user's hand, which is eliminated from the projected image.

Virtual Clay Volflex

Virtual clay is one of the ultimate goals of interactive technique of 3D graphics. Digital tools for 2D paint are mature technology. On the other hand, tools for 3D-shape manipulation are at only preliminary stage. Shape design of 3D objects is one of the major application areas of haptic interface. Shape design of 3D object requires good sensation of haptics.

The Volflex provides an effective interface device for manipulating virtual clay using a lattice of air balloons. Two-dimensional paint tools have been popular, and digital picture is easy to draw. The Volflex is a new digital tool for making 3D shapes. It has the potential to be a revolution in industrial design. Designers use their palm or the joints of their fingers to deform a clay model when carrying out rough-design tasks. The Volflex has the ability to support such natural manipulation.

The Volflex is not only a tool for 3D-shape design, but is also an interactive artwork. Physical properties of the virtual object can be designed by programming the balloons' controllers. A projected image can be also designed. The combination of haptic and visual display provides a new platform for interactive sculpture.

TECHNOLOGIES IN FULL-BODY HAPTICS

Treadmill-Based Locomotion Interface

Basic Design of the Torus Treadmill

A key principle of treadmill-based locomotion interface is to make the floor move in a direction opposite of that of the walker. The motion of the floor cancels displacement of the walker in the real world. The major problem with treadmill-based locomotion interface is that there is not yet a way to allow the walker to change direction. An omni-directional active floor enables a virtually infinite area. In order to realize an infinite walking area, geometric configuration of an active floor must be chosen. A closed surface driven by actuators has an ability to create an unlimited floor. The following requirements for implementation of the closed surface must be considered:

1. The walker and actuators must be put outside the surface.
2. The walking area must be a plane surface.
3. The surface must be made of a material that stretches very little.

The shape of a closed surface, in general, is a surface with holes. If the number of holes is zero, the surface is a sphere. The sphere is the simplest infinite surface. However, the walking area of the sphere is not a plane surface. A very large diameter is required to make a plane surface on a sphere, which restricts implementation of the locomotion interface.

A closed surface with one hole like a doughnut is called *torus*. A torus can be implemented using a group of belts, which make a plane surface for the user to walk on. A closed surface with more than one hole cannot make a plane walking surface. Thus, the torus is the only form suitable for a locomotion interface.

Mechanism and Performance

The Torus Treadmill operates using a group of belts connected to each other. The Torus Treadmill is realized by these belts (Iwata, 1999). Figures 12.16 and 12.17 illustrates basic structure of the Torus Treadmill, which employs 12 treadmills. These treadmills move the walker along in an "X" direction. Twelve treadmills are connected side by side and driven in a perpendicular direction., a motion that moves the walker along a "Y" direction.

Figure 12.18 shows the apparatus. Twelve treadmills are connected to four chains and mounted on four rails. The chain drives the walker along the Y direction. The rail supports the weight of the treadmills and the walker. An AC motor is employed to drive the chains. The power of the motor is 200 W and is controlled by an inverter. The maximum speed of rotation is 1.2 m/s. The maximum acceleration is 1.0 m/s^2. The deceleration caused by friction is 1.5 m/s^2. Frequency characteristics are

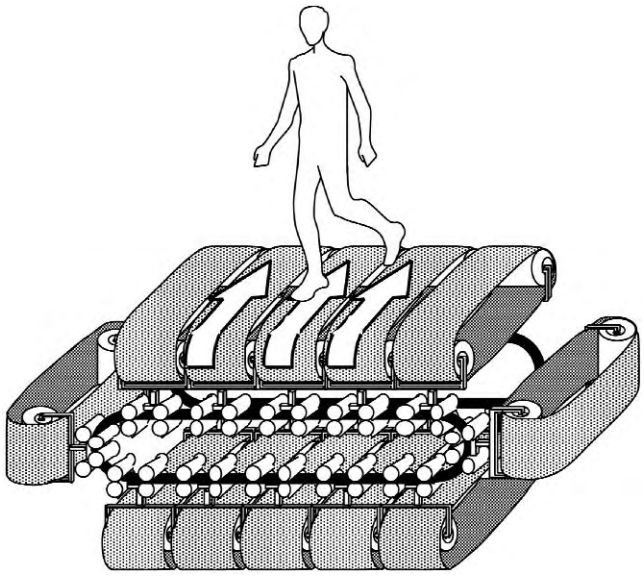


FIGURE 12.16. Structure of Torus Treadmill (X motion).

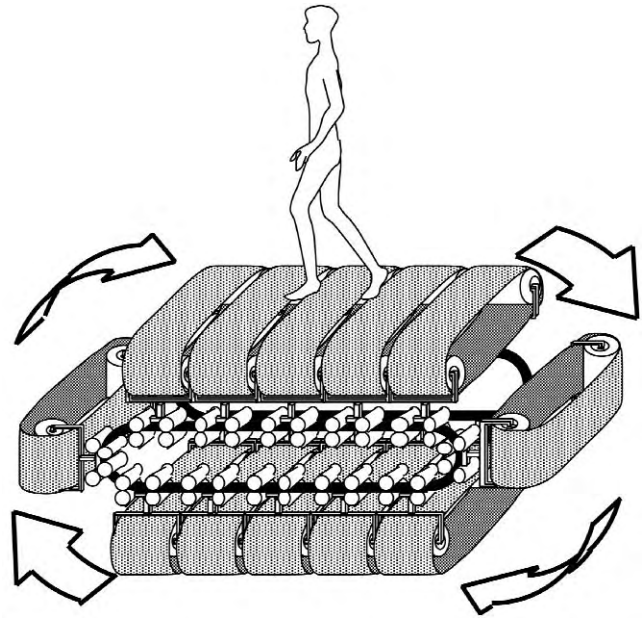


FIGURE 12.17. Structure of Torus Treadmill (Y motion).

limited by a circuit protector of the motor driver. The maximum switching frequency is 0.8 Hz.

Each treadmill is equipped with an AC motor. In order to shorten the length of the treadmill, the motor is put underneath the belt. The power of each motor is 80 W and is controlled by an inverter. The maximum speed of each treadmill is 1.2 m/s. The maximum acceleration is 0.8 m/s^2. The deceleration caused by friction is 1.0 m/s^2. The width of each belt is 250 mm and the overall walkable area is 1m $\times$ 1m.

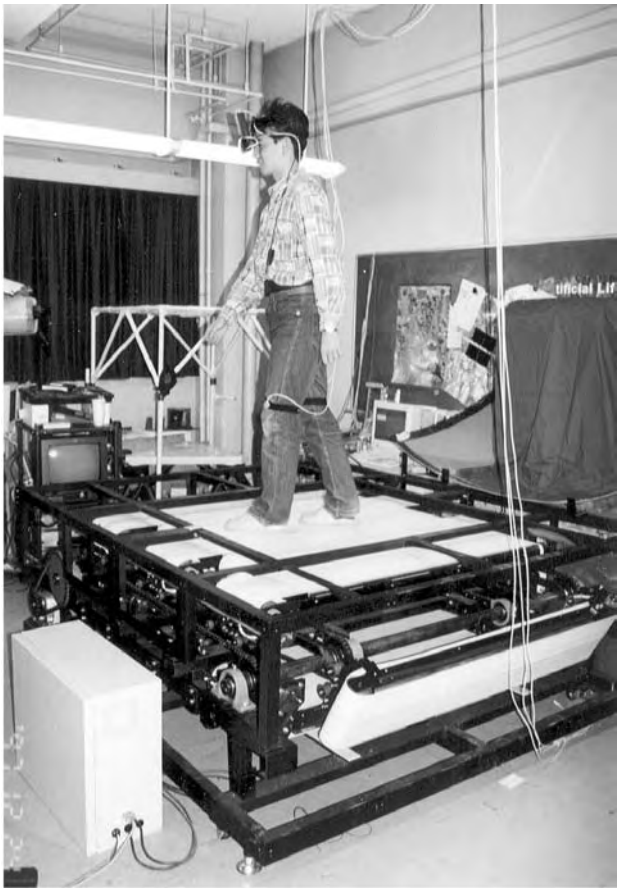


FIGURE 12.18. Torus Treadmill.

A problem with this mechanical configuration is the gap between the belts in the walking area. In order to minimize the gap, we put a driver unit of each treadmill alternatively. The gap is only 2 mm wide in this design.

Control Algorithm of the Torus Treadmill

A scene of the virtual space is generated corresponding with the results of motion tracking of the feet and head. The motion of the feet and head is measured by a Polhemus FASTRACK. The device measures a six-degree-of-freedom motion. The sampling rate of each point is 20 Hz. A receiver is attached to each knee. We cannot put the sensors near the motion floor because a steel frame distorts magnetic field. The length and direction of a step is calculated by the data from those sensors. The user's viewpoint in virtual space moves in correspondence with the length and direction of the steps.

To keep the walker in the center of the walking area, the Torus Treadmill must be driven in correspondence with the walker. A control algorithm is required to achieve safe and natural walking. From our experience in the Virtual Perambulator project, the walker should not be connected to a harness or mechanical linkages, since such devices restrict the motion and inhibit natural walking. The control algorithm of the Torus Tread-

mill must be safe enough to allow removal of the harness from the walker. At the final stage of the Virtual Perambulator Project, we succeeded in removing the harness using a hoop frame. The walker can freely walk and turn around in the hoop, which supports the walker's body while he or she slides his or her feet. We simulated the function of the hoop in the control algorithm of the Torus Treadmill by putting circular deadzone in the center of the walking area. If the walker steps out of the area, the floor moves in the opposite direction, so that the walker is carried back into the deadzone.

Foot-Pad-Based Locomotion Interface

Methods of Presentation of Uneven Surface

One of the major research issues in locomotion interface is the presentation of uneven surface. Locomotion interfaces are often applied for simulation of buildings or urban spaces. Those spaces usually include stairs. A walker should be provided with a sense of climbing up or going down those stairs. In some applications of locomotion interface, such as training simulators or entertainment facilities, rough terrain should be presented.

The presentation of a virtual staircase was tested at the early stage of the Virtual Perambulator project (Iwata & Fujii, 1996). A string was connected to the roller skate of each foot. The string is pulled by a motor. When the walker climbed up a stair, the forward foot was pulled up. When the walker went down the stairs, the backward foot was pulled up. However, this method was not considered successful because of instability.

Later, a 6 DOF motion platform was applied to a final version of the Virtual Perambulator, where a user walks in a hoop frame. The walker stood on the top plate of the motion platform. Pitch and heave motion of the platform were used. When the walker stepped forward to climb a stair, the pitch angle and vertical position of the floor increased. After finishing the climbing motion, the floor went back to the neutral position. When the walker stepped forward to go down a stair, the pitch angle and vertical position of the floor decreases. This inclination of the floor is intended to present height differences between the feet. The heave motion is intended to simulate vertical acceleration. However, this method failed in simulation of stairs, mainly because the floor was flat.

A possible method to create a difference in height between the feet is the application of two large manipulators, such as the BiPort. A 4 DOF manipulator driven by hydraulic actuators is connected to each foot. The major problem with this method is the way in which the manipulators trace the turning motion of the walker. When the walker turns around, two manipulators interfere with each other.

The Torus Treadmill provides a natural turning motion in which the walker can physically turn about on the active floor. This turning motion using the feet is a major contribution to human spatial recognition performance. Vestibular and proprioceptive feedback is essential to the sense of orientation (Iwata & Yoshida, 1999). The Torus Treadmill can be modified for simulation of an uneven surface. If we install an array of linear actuators on each treadmill, an uneven floor can be realized by controlling the length of each linear actuator. However, this method is almost impossible to implement, because a very large number of linear actuators are required to cover the surface of the torus-shaped treadmills and control of the signal for each actuator must be transmitted wirelessly.

Basic Design of the GaitMaster

A new locomotion interface that simulates an omni-directional uneven surface has been designed, which is called the "GaitMaster." The core elements of the device are two 6 DOF motion-bases mounted on a turntable. Figure 12.19 illustrates the basic configuration of the GaitMaster.

A walker stands on the top plate of the motion base. Each motion base is controlled to trace the position of the foot. The turntable is controlled to trace the orientation of the walker. The motion of the turntable removes interference between the two motion bases.

The X and Y motion of the motion base traces the horizontal position of the feet and cancels its motion by moving in the opposite direction. The rotation around the yaw axis traces the horizontal orientation of the feet. The Z motion traces vertical position of the feet and cancels its motion. The rotation around the roll-and-pitch axis simulates the inclination of a virtual surface.

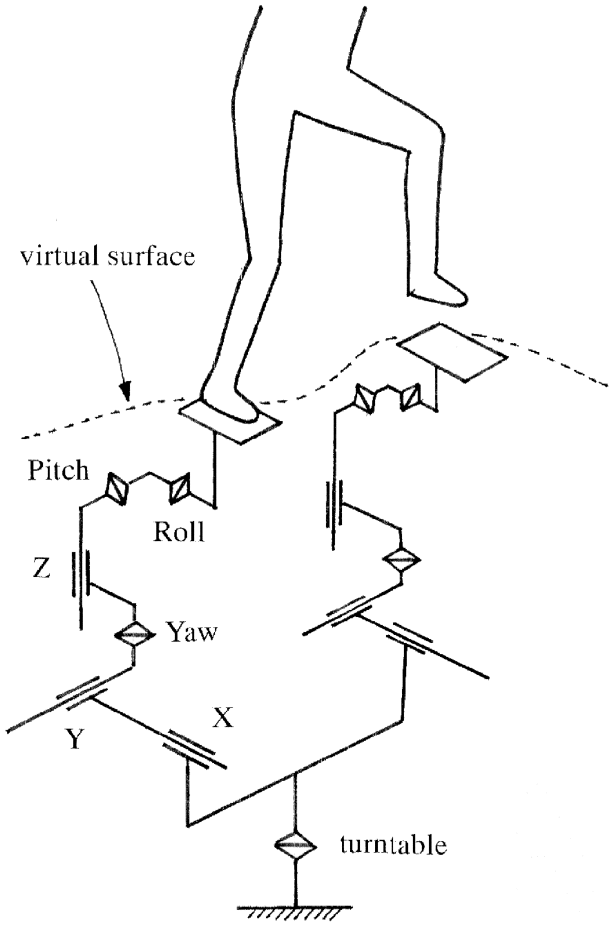


FIGURE 12.19. Basic design of the GaitMaster.

Control Algorithm of the GaitMaster

The control algorithm must keep the position of the walker at the neutral position of the GaitMaster. In order to keep the position maintained, the motion platforms have to cancel the motion of the feet. The principal of the cancellation is as follows:

1. Suppose the right foot is at the forward position and the left foot is at the backward position while walking.
2. When the walker steps forward with the left foot, the weight of the walker is laid on the right foot.
3. The motion platform of the right foot goes backward in accordance with the displacement of the left foot, so that the central position of the walker is maintained.
4. The motion platform of the left foot follows the position of the left foot. When the walker finishes stepping forward, the motion platform supports the left foot.

If the walker climbs up or goes down stairs, a similar procedure can be applied. The vertical motion of the feet is canceled using the same principle. The vertical displacement of the forward foot is canceled in accordance with the motion of the backward foot, so that the central position of the walker is maintained at the neutral height. Figure 12.20 illustrates the method of canceling the climbing-up motion.

The turntable rotates so that the two motion platforms can trace the rotational motion of the walker. If the walker changes the direction of his or her steps, the turntable rotates to trace the walker's orientation. The orientation of the turntable is determined according to direction of the feet. The turntable rotates so that its orientation is at the middle of the feet. The walker can physically turn around on the GaitMaster using this control algorithm of the turntable.

Prototype GaitMaster

Figure 12.21 shows the prototype GaitMaster. In order to simplify the mechanism of the motion platform, the surface of the virtual space was defined as sets of plainer surfaces. Most buildings or urbane spaces can be simulated without inclining of the floor. Thus, we can neglect the roll-and-pitch axis of the motion platforms. Each platform of the prototype GaitMaster is composed of three linear actuators top of which a yaw joint is mounted. We disassembled a 6 DOF Stewart platform and made two XYZ stages. Three linear guides are applied to support the orientation of the top plate of the motion platform. The payload of each motion-platform is approximately 150 Kg. A rotational joint around the yaw axis is mounted on each motion platform. The joint is equipped with a spring that moves the feet in the neutral direction.

A turntable is developed using a large DD motor. The maximum angular velocity is 500 deg/sec. A 3 DOF goniometer is connected to each foot. The goniometer measures back-and-forth and up-and-down motion as well as yaw angle. The control algorithm mentioned in the former section was implemented and succeeded in the presentation of virtual staircases.

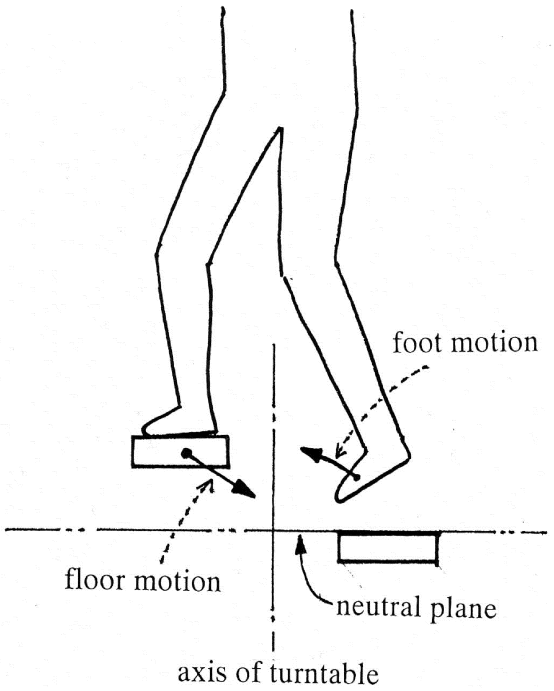


FIGURE 12.20. Canceling the climbing-up motion.



FIGURE 12.21. GaitMaster.

Robot-Tile-Based Locomotion Interface

CirculaFloor Project

Locomotion interfaces often require bulky hardware, since they have to carry the user's whole body. Also, the hardware is not easy to reconfigure to improve its performance or to add new functions. Considering these issues, the goals of the Circula Floor project are,

1. to develop compact hardware for the creation of an infinite surface for walking; The major disadvantage of existing locomotion interfaces is their difficult installation. We need to solve this problem for a demonstration at SIGGRAPH.
2. to develop scalable hardware architecture for future improvement of the system; Another disadvantage of existing locomotion interfaces is the difficulty of improving the system. We have to design a new hardware architecture that allows us to easily upgrade the actuation mechanism and add new mechanisms for the creation of uneven surfaces.

In order to achieve these goals, we designed a new configuration for a locomotion interface by using a set of omni-directional movable tiles. Each tile is equipped with a holonomic mechanism that achieves omni-directional motion. An infinite surface is simulated by the circulation of the movable tiles. The motion of the feet is measured by position sensors. The tile moves opposite to the measured direction of the walker so that the motion of the step is canceled. The position of the walker is fixed in the real world by this computer-controlled motion of the tiles. The circulation of the tiles has the ability to cancel the displacement of the walker in an arbitrary direction. Thus, the walker can freely change direction while walking. Figure 12.22 shows overall view of CirculaFloor

The CirculaFloor is a new method that takes advantage of both the treadmill and footpad. It creates an infinite omni-directional surface using a set of movable tiles. The combination of tiles provides a sufficient area for walking, and thus precision tracing of the foot position is not required. It has the potential to create an uneven surface by mounting an up-and-down mechanism onto each tile.

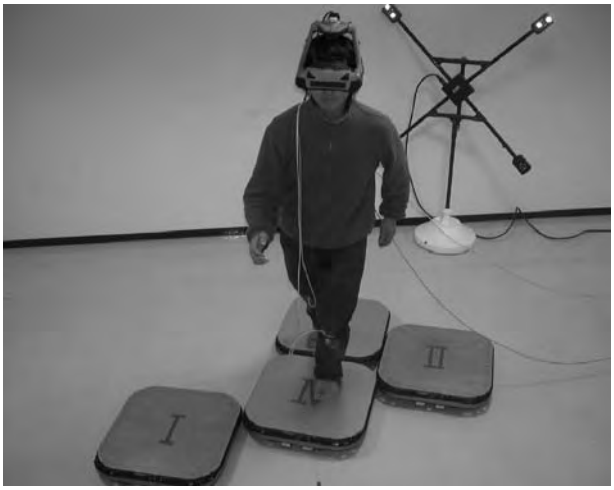


FIGURE 12.22. CirculaFloor.

Method of Creating Infinite Surface

The current method of circulating the movable tiles is designed to satisfy the following conditions: (a) Two of the movable tiles are used for pulling back the user to the center of the dead zone; (b) The rest of the movable tiles are used to create a new front surface; (c) These tiles are moved the shortest distances to the next destination, while they avoid colliding with other tiles; (d) The control program allocates all destinations to the tiles, when the tiles reach their destination; (e) The tiles don't rotate corresponding to walking direction to simplify the algorithm.

Considering the above conditions, the circulation method is varied corresponding to the walking direction. Three modes, "alternating circulation," "unidirectional circulation," and "cross circulation" are designed corresponding to the direction (Fig. 12.23). Representative motion of each mode is illustrated in Fig. 12.24 and 12.25.

Alternating circulation (Fig. 12.24): This mode is adopted for the directions between ±15 deg and ±75–105 deg. The tiles used for creating a new front surface (white-colored tiles in Figure 12.3) move around to the front of the tiles for alterna-

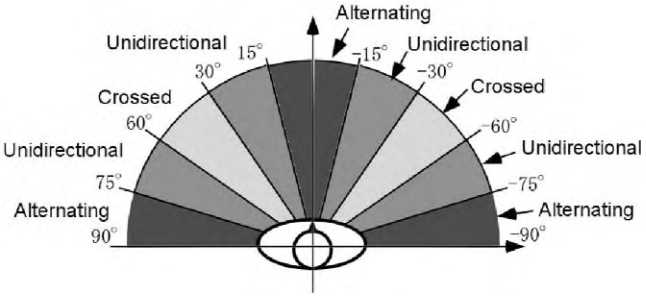


FIGURE 12.23. Pulling-back modes corresponding to walking direction.

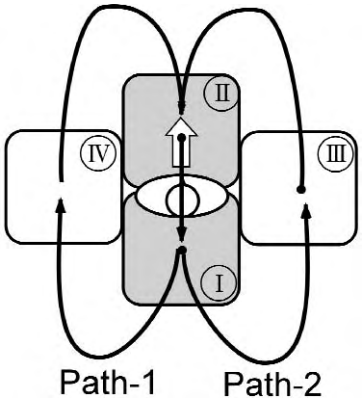


FIGURE 12.24. Circulation of movable tiles in alternating mode.

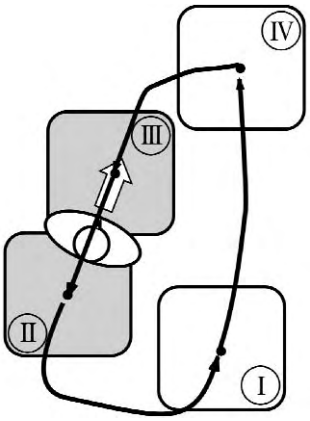


FIGURE 12.25. Circulation of movable tiles in unidirectional mode.

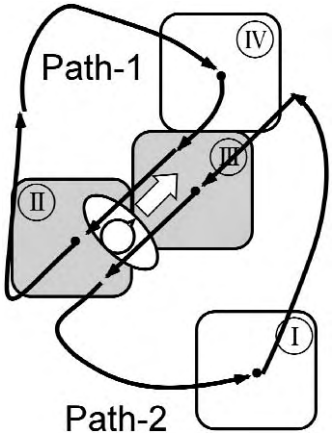


FIGURE 12.26. Circulation of movable tiles in cross mode.

tively pulling back (in Figure 12.24, gray-colored tiles) from left (Path-1)/right (Path-2) sides.

Unidirectional circulation (Fig. 12.25): This mode is adopted for the directions between ±15–30 deg and ±60–75 deg. The tiles used for creating a new front surface move around to the right/left front of the tiles for pulling back with a unidirectional circulation.

Cross-circulation (Fig. 12.26): This mode is adopted for the directions between ±30–60 deg. The tiles used for creating a new front surface move around to the left/right front (Path-1) or the left/right sides (Path-2) of the tiles for pulling back.

When a user of the CirculaFloor switches the walking direction, the control program calculates the nearest phase of each tile by using a template-matching technique corresponding to the new direction. Then the tiles take the shortest way to their destinations.

APPLICATION AREA FOR HAPTIC INTERFACE

Application Area for Finger/Hand Haptics

Medicine. Medical applications for haptic interfaces are currently growing rapidly. Various surgical simulators have been developed using a tool-handling-type force display. We developed a simulator for laparoscopic surgery using the Haptic Master. Simulator software using PHANToM are commercially available.

Palpation is typically used in medical examinations. The FEELEX 2 is designed to be used as a palpation simulator. If we display a virtual tumor based on a CT or an MRI image, a medical doctor can palpate the internal organs before surgery, and this technique can be also applied to telemedicine. Connecting two FEELEXs together via a communication line would allow a doctor to palpate a patient remotely.

3D-shape modeling. The design of 3D shapes definitely requires haptic feedback. A typical application of the tool-handling-type force display is in 3D-shape modeling. One of the most popular applications of the PHANToM system is as a modeling tool. Such a tool-handling-type force display allows a user to point contact, and point-contact manipulation is most suited for precision-modeling tasks. However, it isn't effective when the modeling task requires access to the whole shape. Designers use their palm or the joints of their fingers to deform a clay model when carrying out rough-design tasks. The FEELEX has the ability to support such natural manipulation.

HUI (Haptic User Interface). Today, touchscreens are widely used in automatic teller machines, ticketing machines, information kiosks, and so on. A touchscreen enables an intuitive-user interface, although it lacks haptic feedback. Users can see virtual buttons, but they can't feel them. This is a serious problem for a blind person. The FEELEX provides a barrier-free solution to the touchscreen-based user interface. Figure 12.13 shows an example of a haptic touch-screen using the FEELEX 1.

Art. Interactive art may be one of the best applications of the FEELEX system. As we discussed earlier, the Anomalocaris has been exhibited in a museum in Austria. It succeeded in evoking haptic interaction with many visitors. The FEELEX can be used for interactive sculptures. Visitors are usually prohibited from touching physical sculptures. However, they can not only touch sculptures based around FEELEX, they can also deform them.

Application Areas for Locomotion Interfaces

As a serious application of our locomotion interface, we are working with the Ship Research Laboratory to develop an "evacuation simulator (Yamao et al., 1996)." The Ship Research Laboratory is a national research institute that belongs to the ministry of transportation of Japan. Analysis of evacuation of passengers during maritime accidents is very important for ship safety. However, it is impossible to carry out experiments with human subjects during an actual disaster. Therefore, they introduced virtual-reality tools for simulation of disaster in order to analyze evacuation of passengers. They built a virtual ship that models the generation of smoke and the inclination of the vessel. Experiments of evacuation are carried out for the construc-

tion of mathematical models of passengers' behavior in a disaster. The Torus Treadmill will be effective in such experiments.

Locomotion by walking is intuitive and inevitable in the study of human behavior in virtual environments. We are applying the system to research a human model of evacuation in maritime accidents. The GaitMaster can be applied to other areas than the Torus Treadmill. Its application may include rehabilitation of walking or simulation of mountain climbing.

CONCLUSION AND FUTURE PROSPECTS

This chapter describes major topics of haptic interfaces. A number of methods have been proposed implementing haptic interfaces. Future research in this field will include following two issues: (a) safety issues, and (b) psychology in haptics.

Safety Issues

Safety is an important issue in haptic interface. Inadequate control of actuator may injure the user. The exoskeleton and tool-handling-type-force displays have control problems in their contact surface for the virtual objects. Vibration or unwanted forces can be generated back to the user, which is sometimes dangerous. One of the major advantages of FEELEX is safety. The user of FEELEX doesn't wear any special equipment while the interaction is taking place. The contact surface of the FEELEX is physically generated, so it is free from such control problems.

Locomotion interface has much more important safety issues. The system supports the full body of the user, so that inadequate control causes major damage to the user. Specialized hardware for keeping the walker safe must be developed.

Psychology in Haptics

There have been many findings regarding haptic sensation. Most of these are related to skin sensation, and very few research activities include muscle sensation. Among these, Lederman and Klatzky's work was closely related to the design of the force display (Lederman & Klatzky, 1987). Their latest work involved spatially distributed forces (Lederman & Klatzky, 1999). They performed an experiment involving palpation. The subjects were asked to find a steel ball placed underneath a foam-rubber cover. The results showed that steel balls smaller than 8 mm in diameter decreased the score. This finding supports our specification for the FEELEX 2 in which the distance between rods is 8 mm. This kind of psychological study will support future development of haptic interface.

Haptics is indispensable for human interaction in the real world. However, haptics is not commonly used in the field of HCI. Although there are several commercially avalable haptic interfaces, they are expensive and limited in their fuction. Image display has a history of over 100 years. Today, image displays, such as television and movies, are used in everyday life. On the other hand, haptic interface has only 10-year history. There are hazards to overcome for popular use of haptic interface. However, haptic interface is a new frontier of media technology and will definitly contribute to human life.

References

Asanuma, N., Yokoyama, N., & Shinoda, H. (1999). A method of selective stimulation to epidermal skin receptors for realistic touch feedback. *Proceedings of IEEE Virtual Reality '99* (pp. 274–281).

Brooks, F. P., Jr. (1986). A dynamic graphics system for simulating virtual buildings. *Proceedings of the 1986 Workshop on Interactive 3D Graphics* (pp. 9–21). New York: ACM.

Brooks, F. P., et al. (1990). Project GROPE—Haptic displays for scientific visualization. *Computer Graphics, 24*(4).

Burdea, G. C. (1996). *Force and touch feedback for virtual reality*. Wiley-Interscience.

Burdea, G., Zhuang, J., Roskos, E., Silver, D., & Langlana, L. (1992). A Portable Dextrous Master with Force Feedback. *Presence, 1*(1).

Christensen, R., Hollerbach, J. M., Xu, Y., & Meek, S. (1998). Inertial force feedback for a locomotion interface. Proceedings of ASME Dynamic Systems and Control Division (Vol. 64; pp. 119–126).

Darken, R., Cockayne, W., & Carmein, D. (1997). The Omni-directional treadmill: A locomotion device for virtual worlds. *Proceedings of UIST '97*.

Hirota, K., & Hirose, M. (1996). Simulation and presentation of curved surface in virtual reality environment through surface display. *Proceedings of IEEE VRAIS '96*.

Iwata, H. (1990). Artificial reality with force-feedback: Development of desktop virtual space with compact master manipulator. ACM SIGGRAPH *Computer Graphics, 24*(4).

Iwata, H. (1990). Artificial reality for walking about large scale virtual space. *Human Interface News and Report, 5*(1), 49–52.

Iwata, H. (1993). Pen-based Haptic virtual environment. *Proceedings of IEEE VRAIS '93*.

Iwata, H. (1994). Desktop Force Display. *SIGGRAPH 94 Visual Proceedings*.

Iwata, H. (1999). Walking about virtual space on an infinite floor. *Proceedings of IEEE Virtual Reality '99* (pp. 236–293).

Iwata, H., & Fujii, T. (1996). Virtual perambulator: A novel interface device for locomotion in virtual environment. Proceedings of IEEE 1996 Virtual Reality Annual International Symposium (pp. 60–65).

Iwata, H., & Yoshida,Y. (1999). Path reproduction tests using a torus treadmill. *Presence, 8*(6), 587–597.

Kajimoto, H., Kawakami, N., Maeda, T., & Tachi, S. (1999). Tactile Feeling Display using Functional Electrical Stimulation. *Proceedings of ICAT '99* (pp. 107–114).

Kawai, Y., & Tomita, F. (2000). A support system for the visually impaired to recognize three-dimensional objects. *Technology and Disability, 12*(1), 13–20.

Kontarinis, D. A., & Howe, R .D. (1995). Tactile display of vibratory information in teleoperation and virtual environment. *Presence, 4*(4), 387–402.

Lederman, S. J., & Klatzky, R. L. (1987). Hand movements: A window into haptic object recognition. *Cognitive Psychology, 19*(3), 342–368.

Lederman, S. J., & Klatzky, R. L. (1999). Sensing and displaying spatially distributed fingertip forces in haptic interfaces for teleoperators and virtual environment system. *Presence*, *8*(1).

Lorenzo, M. et al. (1995). OSIRIS. *SIGGRAPH '95 Visual Proceedings* (p. 129).

Massie, T., & Salisbury, K. (1994). The PHANToM Haptic Interface: A device for probing virtual objects. *ASME Winter Anual Meeting* (DSC-Vol.55-1).

McNeely, W. (1993). Robotic graphics: A new approach to force feedback for virtual reality. *Proceedings of IEEE VRAIS '93.*

Minsky, M., & Lederman, S. J. (1997). Simulated Haptic Textures: Roughness. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Proceedings of the ASME Dynamic Systems and Control Division, DSC-Vol. 58.

Moy, G.., Wagner, C., & Fearing, R. S. (2000, April). A compliant tactile display for teletaction. *IEEE International Conference on Robotics and Automation.*

Murakami, T., & Nakajima, N. (1994). Direct and intuitive input device for 3D shape deformation. *ACM CHI, Conference on Human Factors in Computing Systems*, (pp. 465–470).

Noma, H., Sugihara, T., & Miyasato. (2000). Development of ground surface simulator for Tel-E-Merge System. *Proceedings of IEEE Virtual Reality* (pp. 217–224).

Poston, R. et al. (1997). A whole body kinematic display for virtual reality applications. *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 3006–3011).

Prat, David R. et al. (1994). Insertion of an articulated human into a networked virtual environment. *Proceedings of the A I Simulation, and Planning in High Autonomy Systems Conference* (pp. 7–9).

Sinclair, M. (1997) The Haptic lens. *SIGGRAPH '97 Visual Proceedings* (p. 179).

Slater, M. et al. (1994). Steps and ladders in Virtual Reality. Virtual Reality Technology, World Scientific Publication, pp. 45–54 (1994).

Tachi, S. et al. (1994). A construction method of virtual Haptic space. Proceedings of ICAT '94.

Yamao, T., Ishida, S., Ota, S., & Kaneko, F. (1996). Formal safety assesment—Research project on quantification of risk on lives, MSC67/INF.9 IMO information paper, (1996).

13

NONSPEECH AUDITORY OUTPUT

Stephen Brewster
University of Glasgow

Introduction and a Brief History of Nonspeech Sound and HCI **248**
Why Use Nonspeech Sound in HCIs? **248**
Some Advantages of Sound 248
Some Problems with Sound 249
Comparing Speech and Nonspeech Sounds for Interface Design 249
Avoiding Annoyance 250
Perception of Sound **251**
Technology and Sound Production **252**
A Brief Introduction to Sound Synthesis and MIDI . . . 252
The Musical Instrument Digital Interface—MIDI . . . 253
A Brief Introduction to Sampling 254
Sample Rate 254
Sample Size 254
Comparing MIDI Synthesis to Sampling for Auditory Interface Design 254
Three Dimensional Sound 254
Nonspeech Sound Presentation Techniques **255**
Auditory Icons 255
Design Guidelines for Auditory Icons 256
Earcons 256
Comparing Auditory Icons and Earcons 258
The Applications of Auditory Output **259**
Sonic Enhancement of GUIs 259
Sound for Users with Visual Impairments 260
Sonification 261
Sound for Mobile and Ubiquitous Computing 261
Conclusions **263**
References **263**

INTRODUCTION AND A BRIEF HISTORY OF NONSPEECH SOUND AND HCI

Our sense of hearing is very powerful, and we can extract a wealth of information from the pressure waves entering our ears as sound. Sound gives us a continuous, holistic contact with our environments; we hear a rich set of sounds from our interactions with objects close to us, familiar sounds of unseen friends or family nearby, noises of things to avoid, such as traffic, and noises of things to attend to, such as a ringing telephone. Nonspeech sounds (such as music, environmental sounds, and sound effects) give us different types of information than those provided by speech; they can be more general and more ambient where speech is precise and requires more focus (see chapter XXX for more on speech output). Nonspeech sounds complement speech in the same way that visual icons complement text. For example, icons can present information in a small amount of space as compared to text; nonspeech sounds can present information in a small amount of time as compared to speech. There is less research about nonspeech than speech interfaces, and this chapter will show something of where it has been used and of what it is capable.

The combination of visual and auditory feedback at the user interface is a powerful tool for interaction. In everyday life, these primary senses combine to give complementary information about the world. Our visual system gives us detailed information about a small area of focus, whereas our auditory system provides general information from all around, alerting us to things outside our view. Blattner and Dannenberg (1992) discussed some of the advantages of using this approach in multimedia/multimodal computer systems:

> In our interaction with the world around us, we use many senses. Through each sense, we interpret the external world using representations and organizations to accommodate that use. The senses enhance each other in various ways, adding synergies or further informational dimensions. (p. xviii)

These advantages can be brought to the multimodal (or multisensory) human-computer interface by the addition of nonspeech auditory output to standard graphical displays (see chapter 21 for more on multimodal interaction). Whilst directing our visual attention to one task, such as editing a document, we can still monitor the state of others on our machine using sound. Currently, almost all information presented by computers uses the visual sense. This means information can be missed because of visual overload or because the user is not looking in the right place at the right time. A multimodal interface that integrated information output to both senses could capitalize on the interdependence between them and present information in the most efficient way possible.

The classical uses of nonspeech sound can be found in the human factors literature (see McCormick & Sanders, 1982). Here it is used mainly for alarms, warnings, and status information. Buxton (1989) extended these ideas and suggested that encoded messages could be used to present more complex information in sound. This type of auditory feedback will be considered here.

The other main use of nonspeech sound is in music and sound effects for games and other multimedia applications. These kinds of sounds indicate to the user something about what is going on and try to create a mood for the piece (much as music does in film and radio). As Blattner and Dannenberg (1992) said, "Music and other sound in film or drama can be used to communicate aspects of the plot or situation that are not verbalized by the actors. Ancient drama used a chorus and musicians to put the action into its proper setting without interfering with the plot" (p. xix). Work on auditory output for interaction takes this further and uses sound to present information, such as things that the user might not otherwise see or important events that the user might not notice.

The use of sound to convey information in computers is not new. In the early days, programmers used to attach speakers to their computers' busses or program counters (Thimbleby, 1990). The speaker would click each time the program counter was changed. Programmers would get to know the patterns and rhythms of sound and could recognize what the machine was doing. Another everyday example is the sound of a hard disk. Often, users can tell when a save or copy operation has completed by the noise their disks make. This allows them to do other things while waiting for the copy to finish. Nonspeech sound is therefore an important information provider, giving users knowledge about things in their systems that they may not see.

Two important events kick-started the research area of nonspeech auditory output: the first was the special issue of the HCI journal on nonspeech sound, edited by Buxton (1989). This laid the foundations for some of the key work in the area; it included papers by Blattner on earcons (Blattner, Sumikawa, & Greenberg, 1989), Gaver (1989) on auditory icons, and Edwards (1989) on Soundtrack. The second event was the First International Conference on Auditory Display (ICAD'92) held in Santa Fe in 1992 (Kramer, 1994b). For the first time, this meeting brought the main researchers interested in the area together (see http://www.icad.org for the proceedings of the ICAD conferences). Resulting from these two events was a large growth in research in the area during the 1990s that continues today.

The rest of this chapter goes into detail on all aspects of auditory interface design. It presents some of the advantages and disadvantages of using sound at the interface. Then, a brief introduction to psychoacoustics, or the study of the perception of sound, follows. This is followed by an introduction to sound sampling and synthesis techniques needed for auditory interfaces. The next sections describe the main techniques that are used for auditory information presentation and some of the main applications of sound in human-computer interaction. The chapter finishes with some conclusions about the state of research in this area.

WHY USE NONSPEECH SOUND IN HCI?

Some Advantages of Sound

It is advantageous to use sound at the user interface for many reasons.

Vision and hearing are interdependent. Our visual and auditory systems work well together. Our eyes provide high-resolution information around a small area of focus (with peripheral vision extending further). According to Perrott, Sadralobadi, Saberi, and Strybel (1991), humans view the world through a window of 80° laterally and 60° vertically. Within this visual field, focusing capacity is not uniform. The foveal area of the retina (the part with the greatest acuity) subtends an angle of only 2° around the point of fixation. Sounds, on the other hand, can be heard from above, below, in front, or behind the user, but with a much lower resolution. Therefore, our ears tell our eyes where to look: if there is an interesting sound from outside our view we will turn to look at it to get more detailed information.

Superior temporal resolution. As Kramer (1994a) said, "Acute temporal resolution is one of the greatest strengths of the auditory system" (p. 2). In certain cases, reactions to auditory stimuli have been shown to be faster than reactions to visual stimuli (Bly, 1982).

Reduce the overload from large displays. Modern, large, or multiple monitor graphical interfaces use the human visual sense very intensively. This means that we may miss important information because our visual system is overloaded—we have just too much to look at. To stop this overload, information could be displayed in sound so that the load could be shared between senses.

Reduce the amount of information needed on screen. Related to the point above is the problem with information presentation on devices with small visual displays, such as mobile telephones and personal digital assistants (PDAs). These have very small screens that can easily become cluttered. To solve this, some information could be presented in sound to release screen space.

Reduce demands on visual attention. Another issue with mobile devices is that users who are using them on the move cannot devote all of their visual attention to the device—they must look where they are going to avoid uneven surfaces, traffic, pedestrians, and so forth. In this case, visual information may be missed because the user is not looking at the device. If this were played in sound, then the information would be delivered while the user was looking at something else.

The auditory sense is underutilized. The auditory system is very powerful; we can listen to (and some can compose) highly complex musical structures. As Alty (1995) said, "The information contained in a large musical work (say a symphony) is very large . . . The information is highly organized into complex structures and substructures. The potential therefore exists for using music to successfully transmit complex information to a user" (p. 409).

Sound is attention grabbing. Users can choose not to look at something, but it is harder to avoid hearing it. This makes sound very useful for delivering important information.

Some objects or actions within an interface may have a more natural representation in sound. Bly (1982) suggested, "Perception of sound is different to visual perception, sound can offer a different intuitive view of the information it presents . . ." (p. 14). Therefore, sound could allow us to understand information in different ways.

To make computers more usable by visually disabled people. With the development of graphical displays, user interfaces became much harder for visually impaired people to operate. A screen reader (see the section on sounds for users with visual impairments below) cannot easily read this kind of graphical information. Providing information in an auditory form can help solve this problem and allow visually disabled persons to use the facilities available on modern computers.

Some Problems with Sound

Kramer (1994a) suggested some general difficulties with using sound to present information.

Low resolution: Many auditory parameters are not suitable for high-resolution display of quantitative information. For example, when using sound volume, only very few different values can be unambiguously presented (Buxton, Gaver, & Bly, 1991). Vision has a much higher resolution. The same also applies to spatial precision in sound. Under optimal conditions, differences of about 1° can be detected in front of a listener (see section on three dimensional sound; Blauert, 1997). In vision, differences of an angle of two seconds can be detected in the area of greatest acuity in the central visual field.

Presenting absolute data is difficult: Many interfaces that use nonspeech sound to present data do it in a relative way. Users tell if a value is going up or down through the difference between two sounds. It is difficult to present absolute data unless the listener has perfect pitch, which is rare. In vision, a user needs only to look at a number to get an absolute value.

Lack of orthogonality: Changing one attribute of a sound may affect the others. For example, changing the pitch of a note may affect its perceived loudness and vice versa (see the section on perception of sound).

Transience of information: Sound disappears after it has been presented; users must remember the information that the sound contained or use some method of replaying. In vision, the user can easily look back at the display and see the information again. (This is not always the case—think, for example, of an air conditioning system: its sounds continue for long periods and become habituated. Sounds often continue in the background and only become apparent when they change in some way.)

Annoyance due to auditory feedback: There is one problem with sound that has not yet been mentioned, but it is the one most commonly brought up against the use of sound in user interfaces: annoyance. As this is an important topic, it will be discussed in detail in a later section.

Comparing Speech and Nonspeech Sounds for Interface Design

One obvious question is why not just use speech for output? Why do we need to use nonspeech sounds? Many of the advan-

tages presented above apply to speech as well as nonspeech sounds. However, there are some advantages to nonspeech sounds. If we think of a visual analogy, speech output is like the text on a visual display and nonspeech sounds are like the icons. Presenting information in speech is slow because of its serial nature; to assimilate information, the user must typically hear it from beginning to end, and many words may have to be comprehended before a message can be understood. With nonspeech sounds, the messages are shorter and therefore can be heard more rapidly (although the user might have to learn the meaning of the nonspeech sound whereas the meaning is contained within the speech and so requires no learning—just like the visual case). Speech suffers from many of the same problems as text in text-based computer systems, as this is also a serial medium. Barker and Manji (1989) claimed that an important limitation of text is its lack of expressive capability: It may take many words to describe something fairly simple. Graphical displays that speeded up interactions were introduced, as users could see a picture of the application they wanted instead of having to read its name from a list (Barker & Manji, 1989). In the same way, an encoded sound message can communicate its information in fewer sounds. The user hears the sound then recalls its meaning rather than having the meaning described in words. Increasing the presentation rate of synthetic speech decreases the accuracy of recognition of what is being said. Pictorial icons can also be universal: they can mean the same thing in many different languages and cultures, and the nonspeech sounds have similar universality (given some of the different musical cultures).

An important ability of the auditory system is habituation, where continuous sounds can fade into the background of consciousness after a short period. If the sound changes (or stops), then it would come to the foreground of attention because the auditory system is sensitive to change. Habituation is difficult to achieve with speech because of the large dynamic range it uses. According to Patterson (1982), "The vowels of speech are often 30 dB more intense than the consonants, and so, if a voice warning were attenuated to produce a background version with the correct vowel level the consonants would be near or below masked threshold" (p. 11). It is easier to habituate certain types of nonspeech sounds (think of an air conditioner where you only notice that it was on when it switches off) and sounds can be designed to facilitate habituation if required.

Baddeley (1997) gave evidence to show that background speech, even at low intensities, is much more disruptive than nonspeech sound when recalling information. He reports the unattended speech effect. Unattended speech (e.g., in the background) causes information to be knocked out of short-term memory, whereas noise or nonspeech sound does not. This problem is unaffected by the intensity of the speech, provided that it is audible. This shows a problem for speech at the interface, as it is likely to prove disruptive for other users in the same environment unless it is kept at a very low intensity and, as we saw above, this can cause problems with the ability to hear consonants.

Nonspeech sounds are also good for presenting continuous information, such as a graph of stock market data. In speech, particular values can be spoken out at particular times, but there would be no way to monitor the overall trend of the data. Methods for doing this in nonspeech sounds were developed over 20 years ago (Mansur, Blattner, & Joy, 1985) and have proved to be very effective (see Sound Graphs in perception of sound section below for more on this).

This discussion has shown that there are many reasons to think of using nonspeech sounds in addition to speech in HCIs. Few interfaces make good use of both. Speech in general is good for giving instructions and absolute values; nonspeech sounds are good for giving rapid feedback on actions, quickly presenting highly structured information, and presenting continuous data. Together, they make a very effective means of presenting information nonvisually.

Avoiding Annoyance

The main concern potential users of all auditory interfaces have is annoyance due to sound pollution. There are two aspects to annoyance: A sound may be annoying to the user whose machine is making the noise (the primary user), and it may be annoying to others in the same environment who overhear it (secondary users). Buxton (1989) discussed some of the problems of sound and suggested that some sounds help us (information) and some impede us (noise). We therefore need to design sounds that are more informative and less noise. Of course, one person's informative sounds are another's noise, so it is important to make sure that the sounds on one computer are not annoying for colleagues working nearby.

Few studies focus on the problems of annoyance due to nonspeech sounds in computers. There are, however, many studies of annoyance from speech (e.g., Berglund, Harder, & Preis, 1994), from the sounds of aircraft, traffic, and other environmental noises, and most of these suggest that the primary reason for the annoyance of sound is excessive volume. In a different context, Patterson (1989) investigated some of the problems with auditory warnings in aircraft cockpits. Many of the warnings were added in a "better-safe-than-sorry" manner that led to them being so loud that the pilot's first response was to try to turn them off rather than deal with the problem being indicated. One of Patterson's main recommendations was that the volume of the warnings should be reduced.

A loud sound grabs the attention of the primary user, even when the sound is communicating an unimportant event. As the sound is loud, it travels from one machine to the ears of other people working nearby, increasing the noise in their environments.

So, how can annoyance be avoided? One key way is to avoid using intensity as a cue in sound design for auditory interfaces. Quiet sounds are less annoying. Listeners are also not good at making absolute intensity judgments (see the next section). Therefore, intensity is not a good cue for differentiating sounds anyway.

Headphones can be used so that sounds are heard only by the primary user. This may be fine for users of mobile telephones and music players, but is not always a good solution for desktop users who do not want to be cut-off from their colleagues. Manipulating sound parameters other than intensity can make sounds attention grabbing (but not annoying).

Rhythm or pitch can be used to make sounds demanding because the human auditory system is very good at detecting changing stimuli (see Edworthy, Loxley, & Dennis, 1991). Therefore, if care is taken with the design of sounds in an interface, specifically avoiding the use of volume changes to cue the user, then many of the problems of annoyance can be avoided.

PERCEPTION OF SOUND

This section provides some basic information about the perception of sound that is applicable to nonspeech auditory output. The auditory interface designer must be conscious of the effects of psychoacoustics, or the perception of sound, when designing sounds for the interface. As Frysinger (1990) said, "The characterization of human hearing is essential to auditory data representation because it defines the limits within which auditory display designs must operate if they are to be effective" (p. 31). Using sounds without regard for psychoacoustics may lead to the user being unable to differentiate one sound from another, unable to hear the sounds, or unable to remember them (see Moore, 2003).

What are sounds? Sounds are pressure variations that propagate in an elastic medium (in this case, the air). The pressure variations originate from the motion or vibration of objects. These pressure variations hit the listener's ear and start the process of perceiving the sound. The pressure variations plotted against time can be seen in Fig. 13.1. This shows the simplest form of sound: a sine wave (which might be produced by a tuning fork). A sound is made up from three basic components. Frequency is the number of times per second the wave repeats itself (Fig. 13.1 shows three cycles). It is normally measured in Hertz (Hz). Amplitude is the deviation away from the mean pressure level, or force per unit area of a sound. It is normally measured in decibels (dB). Phase is the position of the start of the wave on the time axis (measured in milliseconds).

Sounds from the real world are normally much more complex than Fig. 13.1 and tend to be made up of many sine waves with different frequencies, amplitudes, and phases. Figure 13.2 shows a more complex sound made of three sine wave components (or partials) and the resulting waveform. Fourier analysis allows a sound to be broken down into its component sine waves (Gelfand, 1981).

The sounds in Fig. 13.1 and Fig. 13.2 are periodic—they repeat regularly over time. This is very common for many types of musical instruments that might be used in an auditory interface. Many natural, everyday sounds (such as impact sounds) are not periodic and do not repeat. The sound in Fig. 13.2 is also harmonic—its partials are integer multiples of the lowest (or fundamental) frequency. This is again common for musical instruments but not for everyday sounds. Periodic harmonic sounds have a recognizable pitch, where as nonperiodic, inharmonic sounds tend to have no clear pitch.

The attributes of sound described above are the physical aspects. There is a corresponding set of perceptual attributes. Pitch is the perceived frequency of a sound. Pitch is roughly a logarithmic function of frequency. It can be defined as the attribute of auditory sensation in terms of which sounds may be ordered on a musical scale (Moore, 2003). In the western musical system, there are 96 different pitches arranged into eight octaves of 12 notes. Tones separated by an octave have the frequency ratio two to one. For example, middle C is 261.63 Hz, the octave above this is at 523.25 Hz, and the octave below at 130.81 Hz. It is one of the most useful and easily controlled aspects of sound and is very useful for auditory interface designers. However, as Buxton et al. (1991) said, "It is important to be aware of the myriad interactions between pitch and other attributes of sound . . ." (p. 2.10). For example, pitch is affected by sound intensity: at less than 2 kHz an increase in intensity increases the perceived pitch, at 3 kHz and over an increase in intensity decreases the perceived pitch (Gelfand, 1981).

Humans can perceive a wide range of frequencies. The maximum range we can hear is from 20 Hz to 20 kHz. This decreases with age, so that at age 70, a listener might only hear a maximum of 10 kHz. It is therefore important to ensure that the sounds in an auditory interface are perceivable by its users (also

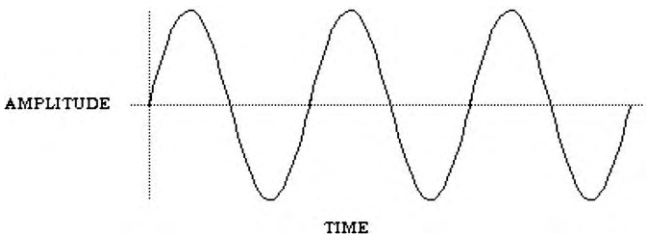


FIGURE 13.1. A sine wave.

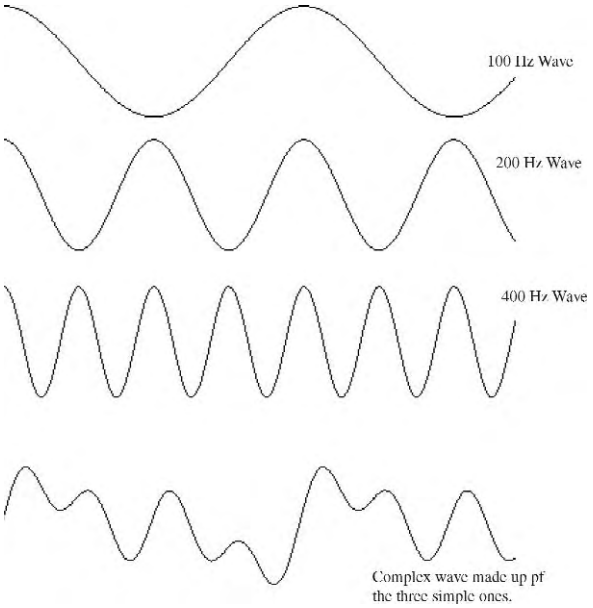


FIGURE 13.2. A complex wave made up of three components with its fundamental at 100 Hz.

poor quality loudspeakers may not be able to cope with the highest or lowest frequencies). Listeners are not good at making absolute judgments of pitch (Moore, 2003). Only 1% of the population has perfect pitch. Another important factor is tone deafness. Moore suggested that this is a misnomer and almost everyone is able to tell that two sounds are different; they are not always able to say which is higher or lower. This can often be overcome with practice, but it is important for the auditory interface designer to be aware of the problem. Mansur et al. (1985) gave evidence of one other important effect: "There appears to be a natural tendency, even in infants, to perceive a pitch that is higher in frequency to be coming from a source that is vertically higher in space when compared to some lower tone" (p. 171). This is important when creating an auditory interface as it could be used to give objects a spatial position. If only stereo position is available to provide spatial cues in the horizontal plane, then pitch could provide them in the vertical plane. Guidelines for the use of pitch (and the other parameters below) are described in the section on Earcons.

Loudness is the perceived intensity of a sound. Loudness (L) is related to intensity (I) according to the *Power Law*: $L = k I^{0.3}$ (Gelfand, 1981). Therefore, a 10dB increase in intensity doubles the perceived loudness of a sound. Loudness is again affected by the other parameters of sound. For example, sounds of between 1 kHz and 5 kHz sound louder at the same intensity level than those outside that frequency range. Humans can perceive a very wide range of intensities: the most intense sound that a listener can hear is 120 dB louder than the quietest. This equates to a ratio of 1,000,000,000,000:1 (Moore, 2003). Buxton et al. (1991) also reported that listeners are "very bad at making absolute judgments about loudness" (p. 2.10) and also that "our ability to make relative judgments of loudness are limited to a scale of about three different levels" (p. 2.11). It is also a primary cause of annoyance (see the section on avoiding annoyance above), so it should be used sparingly by auditory interface designers.

Timbre is the quality of the sound. This attribute of auditory sensation allows a listener to judge two sounds with the same loudness and pitch as dissimilar. It is what makes a violin sound different from a piano, even if both are playing at the same pitch and at the same loudness. Its structure and dimensions are not yet fully understood. It is based partly on the spectrum and dynamics of a sound. Its structure is not well understood, but it is one of the most important attributes of sound that an interface designer can use. As Blattner et al. (1989) said, "Even though timbre is difficult to describe and notate precisely, it is one of the most immediate and easily recognizable characteristics of sound" (p. 26) (both auditory icons and earcons use timbre as one of their fundamental attributes; see section on nonspeech sound presentation techniques below). Many of the synthesis techniques in the next section make it easy for a designer to create and use different timbres.

Duration is another important attribute of sound. Sounds of different durations are used to form rhythmic structures that are a fundamental part of music. Duration can also affect the other parameters of sound. For example, for sounds of less than one second, loudness increases with duration. This is important in auditory interfaces because short sounds are often needed so that the auditory feedback can keep pace with the interactions taking place; accordingly, they must be made loud enough for listeners to hear.

Direction is the position of the sound source. This is often overlooked but is an important aspect of sound in our everyday lives. As mentioned above, one of the key differences between sight and hearing is that sounds can be heard from all around the listener. If a sound source is located to one side of the head, then the sound reaching the further ear will be reduced in intensity (Interaural Intensity Difference; IID) and delayed in time (Interaural Time Difference; ITD; Blauert, 1997). These two factors are key in allowing a listener to localize a sound in space. Humans can detect small changes in the position of a sound source. The minimum auditory angle (MAA) is the smallest separation between two sources that can be reliably detected. Strybel, Manligas and Perrott (1992) reported that in the median plane sound sources only 1° apart can be detected. At 90° azimuth (directly opposite one ear), accuracy falls to ± 10°. This has important implications for auditory displays that use sound position as a cue because higher-resolution placement can be used when sounds are presented in front of the user (see section earcons for more on sound positioning).

TECHNOLOGY AND SOUND PRODUCTION

Most desktop PCs, handheld computers, and mobile telephones have sophisticated sound hardware available for the auditory interface designer. This is normally used for playing games, but it is sufficient to do most of the things required by an auditory interface. The aim of this section is to describe briefly some of the main technologies that are important for designers to understand when creating interfaces.

There are two main aspects to sound production: the first is sound synthesis, and the second is sound sampling and playback. A basic overview will be given focusing on aspects related to audio interfaces. (For much more detail on sound synthesis and MIDI see Roads, 1996; Miranda, 1998; for more on sampling, see Pohlmann, 2005.)

There are many tools available for synthesis and sample playback, and devices such as desktop PCs and mobile telephones have the necessary processing power. The Java programming language (http://www.java.sun.com), for example, has built-in support for a range of synthesis and sampling techniques for many different platforms. Libraries such as Fmod (http://www.fmod.org) allow standard cross-platform sound and work on many different devices and programming languages. All current desktop operating systems provide support for synthesis and sampling. The basic technologies necessary to make the sounds needed for auditory interfaces are thus readily available, but it is important to know something of how they work to utilize them most effectively.

A Brief Introduction to Sound Synthesis and MIDI

The aim of sound synthesis is to generate a sound from a stored model, often a model of a musical instrument. For auditory interfaces, we need a wide, good-quality range of sounds that we

can generate in real time as the user interacts with the interface. Synthesizers come in three main forms: soundcards on PCs, external hardware devices, and software synthesizers.

Most synthesizers are polyphonic, such as able to play multiple notes at the same time (as opposed to monophonic). This is important for auditory interface design, as you might well want to play a chord made up of several notes. Most synthesizers are multitimbral, such as able to play multiple different instruments at the same time. This is again important, as in many situations, a sound composed of two different instruments might be required. The main forms of synthesis will now be briefly reviewed.

Wavetable synthesis is one of the most common and low cost synthesis techniques. Many of the most popular PC soundcards use it (such as the SoundBlaster™ series from Creative Technology—http://www.creative.com). The idea behind wavetable synthesis is to use existing sound recordings (which are often very difficult to synthesize exactly) as the starting point and to create very convincing simulations of acoustical instruments based on them (Miranda, 1998; Roads, 1996). A sample (recording) of a particular sound will be stored in the soundcard. It can then be played back to produce a sound. The sample memory in these systems contains a large number of sampled sound segments, and they can be thought of as a table of sound waveforms that may be looked up and utilized when needed. Wavetable synthesizers employ a variety of different techniques, such as sample looping, pitch shifting, mathematical interpolation, and polyphonic digital filtering, to reduce the amount of memory required to store the sound samples or to get more types of sounds. More sophisticated synthesizers contain more wavetables (perhaps one or more for the initial attack part of a sound and then more for the sustain part of the sound and then more for the final decay and release parts). Generally, the more wavetables that are used the better the quality of the synthesis, but this does require more storage. It is also possible to combine multiple, separately controlled wavetables to create a new instrument.

Wavetable synthesis is not so good if you want to create new timbres, as it lacks some of the flexibility of the other techniques below. Most wavetable synthesizers contain many sounds (often hundreds), so there may not be a great need to create new ones. For most auditory interfaces, the sounds from a good quality wavetable synthesizer will be perfectly acceptable. For desktop computers the storage of large wavetables is no problem, but for mobile telephones with less storage, there may be a much smaller, lower quality set, so care may be needed to design appropriate sounds.

FM (frequency modulation) Synthesis techniques generally use one periodic signal (the modulator) to modulate the frequency of another signal (the carrier; Chowning, 1975). If the modulating signal is in the audible range, then the result will be a significant change in the timbre of the carrier signal. Each FM voice requires a minimum of two signal generators. Sophisticated FM systems may use four or six operators per voice, and the operators may have adjustable envelopes that allow adjustment of the attack and decay rates of the signal. FM synthesis is cheap and easy to implement, and it can be useful for creating expressive new synthesized sounds. However, if the goal is to recreate the sound of an existing instrument, then FM synthesis is not the best choice, as it can generally be done more easily and accurately with wavetable techniques.

Additive (and Subtractive) Synthesis is the oldest form of synthesis (Roads, 1996). Multiple sine waves are added together to produce a more complex output sound (subtractive synthesis is the opposite: a complex sound has frequencies filtered out to create the sound required). It is theoretically possible to create any sound using this method (as all complex sounds can be decomposed into sets of sine waves by Fourier analysis). However, it can be very difficult to create any particular sound. Computer musicians often use this technique, as it is very flexible and easy to create new and unusual sounds, but it may be less useful for general auditory interface design.

Physical Modeling Synthesis uses mathematical models of the physical acoustic properties of instruments and objects. Equations describe the mechanical and acoustic behavior of an instrument. The better the simulation of the instrument, the more realistic the sound produced. Nonexistent instruments can also be modeled and made to produce sounds. Physical modeling is an extremely good choice for synthesis of many classical instruments, especially those of the woodwind and brass families. The downside is that it can require large amounts of processing power, which limits the polyphony.

The Musical Instrument Digital Interface—MIDI

The Musical Instrument Digital Interface (MIDI) allows the real-time control of electronic instruments (such as synthesizers, samplers, drum machines, etc.) and is now very widely used (http://www.midi.org). It specifies a hardware interconnection scheme, a method for data communications, and a grammar for encoding musical performance information (Roads, 1996). For auditory interface designers, the most important part of MIDI is the performance data, which is a very efficient method of representing sounds. Most soundcards support MIDI with an internal synthesizer and provide a MIDI interface to connect to external devices. Both Apple Mac OS and Microsoft Windows have good support for MIDI. Most programming languages now come with libraries supporting MIDI commands.

MIDI performance information is like a piano roll: notes are set to turn on or off and play different instruments over time. A MIDI message is an instruction that controls some aspect of the performance of an instrument. A MIDI message is made up of a status byte, which indicates the type of the message, followed by up to two data bytes that give the parameters. For example, the Note On command takes two parameters: one value giving the pitch of the note required and the other the volume. This makes it a very compact form of presentation.

Performance data can be created dynamically from program code or by a sequencer. In an auditory interface, the designer might assign a particular note to a particular interface event—for example, a click on a button. When the user clicks on the button, a MIDI Note On event will be fired. When the user releases the button, the corresponding Note Off event will be sent. This is a very simple and straightforward way of adding sounds. With a sequencer, data can be entered using classical music notation by dragging and dropping notes onto a stave or by using an external piano-style keyboard. This could then be

saved to a MIDI file for later playback (or could be recorded and played back as a sample—see comparing MIDI synthesis to sampling for auditory interface design section below).

A Brief Introduction to Sampling

In many ways, sampling is simpler than synthesis. The aim is to make a digital recording of an analogue sound and then to be able to play it back later, with the played back sound matching the original as closely as possible. There are two important aspects: sample rate and sample size.

Sample Rate

Sample rate is the number of discrete snapshots of the sound that are taken, often measured per second. The higher the sampling rate is, the higher the quality of the sound when it is played back. With a low sampling rate, few snapshots of the sound are taken, and the recording will not match well the sound being recorded. The Sampling Theorem (Roads, 1996) states that to reconstruct a signal, the sampling frequency must be at least twice the frequency of the signal being sampled. As mentioned above, the limit of human hearing is around 20 kHz, therefore a maximum rate of 40 kHz is required to be able to record any sound that a human can hear. The standard audio CD format uses a sample rate of 44.1 kHz, meaning that is covers all of the frequencies that a human can hear. If a lower sampling rate is used, then higher frequencies are lost. For example, the .au sample format uses a sampling rate of 8 kHz, meaning that only frequencies of less than 4 kHz can be recorded (see Bagwell, 1998).

Higher sampling rates generate much more data than lower ones, so they may not always be suitable if storage is limited (for example, on a mobile computing device). An auditory interface designer must think about the frequency range of the sounds needed in an interface. This might allow the sample rate to be reduced. If the highest quality is required, you must be prepared to deal with large audio files.

Sample Size

The larger the sample size is, the better the quality of the recording, as more information is stored at each snapshot of the sound. Sample size defines the volume (or dynamic) range of the sound. With an 8-bit sample, only 256 discrete amplitude (or quantization) levels can be represented. To fit an analogue sound into one of these levels might cause it to be rounded up or down, and this can add noise to the recording. CD quality sounds use 16-bit samples, giving 65536 different levels, so the effects of quantization are reduced. Many high quality samplers use 24-bit samples to reduce the problems of quantization further.

The two main bit sizes used in most soundcards are 8 and 16 bits. As with sample rates, the main issue is size: 16-bit samples require a lot of storage, especially at high sample rates. Audio CD quality sound generates around 10 Mbytes of data per minute. Compression techniques such as MP3 can help reduce the amount of storage but keep quality high.

Comparing MIDI Synthesis to Sampling for Auditory Interface Design

MIDI is very flexible, as synthesizers can generate sound in real time, as it is needed. If you do not know all of the sounds you might want in your auditory interface in advance, then this can very effective—as the sound is needed, it is just played by the synthesizer. A system that is based around samples can only play back samples that have been prerecorded and stored.

Another advantage of MIDI is that sounds can be changed. Once a sample has been stored, it is difficult to change it. For example, it is possible to change the speed and pitch of a sound independently with MIDI. If a sample is played back at a different speed its pitch will change, which may cause undesirable effects.

MIDI commands are also very small; each command might only take up two or three bytes. Generating sounds from code in your auditory interface is straightforward. For instance, files containing high quality stereo sampled audio require about 10 Mbytes of data per minute of sound, while a typical MIDI sequence might consume less than 10 Kbytes of data per minute of sound. This is because the MIDI file does not contain the sampled audio data; it contains only the instructions needed by a synthesizer to play the sounds.

Samples have the advantage that you, as the interface designer, know exactly what your sound will sound like. With MIDI, you are at the mercy of the synthesizer on the user's machine. It may be of very poor quality, and the sounds might not sound anything like the ones you designed. With samples, all of the information about the sound is stored so that you can guarantee it will sound like the recording you made (given the possible limitations of the speakers on the user's machine).

It is also not possible to synthesize all sounds. Much of the work in sound synthesis has focused on synthesizing musical instruments. Few synthesizers can do a good job of creating natural, everyday sounds. If you want to use natural sounds in your interface, you are limited to using samples.

Three Dimensional Sound

Much of the recorded sound we hear is in stereo. A stereo recording uses differences in intensity between the ears. From these differences, the listener can gain a sense of movement and position of a sound source in the stereo field. The perceived position is along a line between the two loudspeakers or, if they are wearing headphones, inside the head between the listeners' ears. This simple, inexpensive technique can give useful spatial cues at the auditory interface. This is being taken further to make sounds appear as coming from around a user (in virtual 3D) when only a small number of loudspeakers (or even just a pair of headphones) are used. Spatial sound can be used for a range of things, including giving directional information, spreading sound sources around the head to help users differentiate

simultaneous sounds, and creating audio windows in which to present information (see the section on sound for mobile and ubiquitous computing).

As well as the ITD and IID, in the real world we use our pinnae (the outer ear) to filter the sounds coming from different directions so that we know where they are coming from. To simulate sounds as coming from around the user and outside of the head when wearing headphones, sounds entering the ear are recorded by putting microphones into the ear canals of listeners. The differences between the sound at the sound source and at the ear canal are then calculated and the differences, or head-related transfer functions (HRTFs), derived are used to create filters with which stimuli can be synthesized (Blauert, 1997). This research is important, as three-dimensional auditory interfaces that are more natural can be created, with sounds presented around the user, as they would be in real life. Almost all current PC soundcards can generate such 3D sounds, as they are often used in games.

The main problem with providing simulated 3D sound through headphones comes from the general HRTFs used. If your ears are not like the ears of the head (often a dummy head) from which the HRTFs were generated, then you are likely to feel that the sounds are coming from inside your head and not outside. It is also very easy to confuse front and back so that listeners cannot tell if a sound is in front or behind. Vertical positioning is also difficult to do reliably. This means that many designers who use spatial sound in their interfaces often limit themselves to a plane cutting through the head horizontally at the level of the ears, creating a 2.5D space. This reduces the space in which sounds can be presented but avoids many of the problems of users not being able to localize the sounds properly.

To improve quality, head tracking is often used. Once the orientation of the user's head is known, sounds can be respatialized to remain in position when the head turns. Active listening is used to disambiguate the location of a sound—listeners naturally make small head movements and these change the IID and ITD, cueing the listener to the location of the sound. Using such tracking can significantly improve the performance of 3D auditory user interfaces (Marentakis & Brewster, 2004). Marentakis and Brewster also showed that targets should be around $\pm 10°$ in size when head tracking is used to enable accurate localization and selection.

NONSPEECH SOUND PRESENTATION TECHNIQUES

The two main types of nonspeech audio presentation techniques commonly used are auditory icons and earcons. Substantial research has gone into developing both of these, and the main work is reviewed below.

Auditory Icons

Gaver (1989, 1997) developed the idea of auditory icons. These natural, everyday sounds can be used to represent actions and objects within an interface. Gaver defined them as "everyday sounds mapped to computer events by analogy with everyday sound-producing events. Auditory icons are like sound effects for computers" (p. 68). Auditory icons rely on an analogy between the everyday world and the model world of the computer (Gaver, 1997; for more examples of the use of earcons see the work on Mercator and Audio Aura described in sections on sounds for users with visual impairments and sound for mobile and ubiquitous computing).

Gaver (1997) used sounds of events that are recorded from the natural environment, such as tapping or smashing sounds. He used an ecological listening approach (Neuhoff, 2004), suggesting that people do not listen to the pitch and timbre of sounds but to the sources that created them. When pouring liquid, a listener hears the fullness of the receptacle, not the increases in pitch. Another important property of everyday sounds is that they can convey multidimensional data. When a door slams, a listener may hear the size and material of the door, the force that was used, and the size of room on which it was slammed. This could be used within an interface so that selection of an object makes a tapping sound, the type of material could represent the type of object, and the size of the tapped object could represent the size of the object within the interface.

Gaver (1989) used these ideas to create auditory icons and from these built the SonicFinder. This ran on the Apple Macintosh and provided auditory representations of some objects and actions within the interface. Files were given a wooden sound, applications a metal sound, and folders a paper sound. The larger the object the deeper the sound it made. Thus, selecting an application meant tapping it—it made a metal sound that confirmed that it was an application and the deepness of the sound indicated its size. Copying used the idea of pouring liquid into a receptacle. The rising of the pitch indicated that the receptacle was getting fuller and the copy progressing.

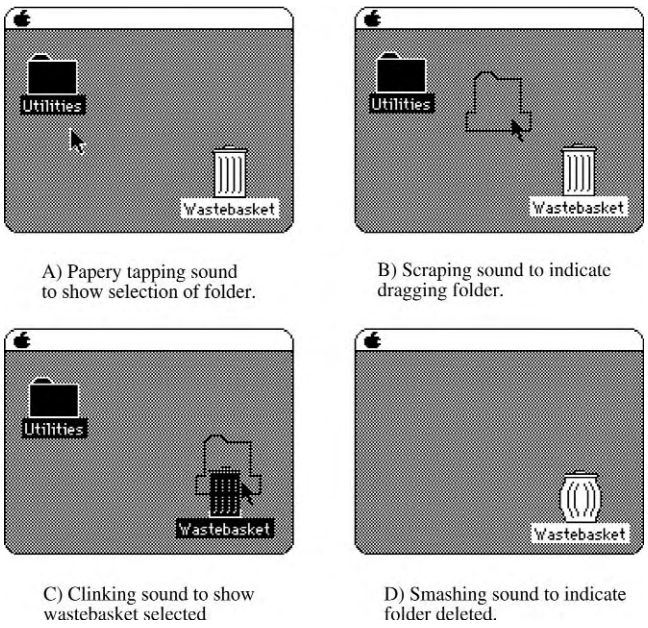


FIGURE 13.3. An interaction showing the deletion of a folder in the SonicFinder (from Gaver, 1989).

To demonstrate how the SonicFinder worked a simple interaction is provided in Fig. 13.3, showing the deletion of a folder. In (a), a folder is selected by tapping on it; this causes a papery sound, indicating that the target is a folder. In (b), the folder is dragged toward the wastebasket, causing a scraping sound. In (c), the wastebasket becomes highlighted, and a clinking sound occurs when the pointer reaches it. Finally, in (d), the folder is dropped into the wastebasket, and a smashing sound occurs to indicate it has been deleted (the wastebasket becomes fat to indicate there is something in it).

Problems can occur with representational systems, such as auditory icons, because some abstract interface actions and objects have no obvious representation in everyday sound. Gaver (1989) used a pouring sound to indicate copying because there was no natural equivalent; this is more like a sound effect. He suggested the use of movie-like sound effects to create sounds for things with no easy representation. This may cause problems if the sounds are not chosen correctly, as they will become more abstract than representational and the advantages of auditory icons will be lost.

Gaver, Smith, and O'Shea (1991) developed the ideas from the SonicFinder further in the ARKola system, which modeled a soft drink factory. The simulation consisted of a set of nine machines split into two groups: those for input, and those for output. The input machines supplied the raw materials; the output machines capped the bottles and sent them for shipping. Each machine had an on/off switch and a rate control. The aim of the simulation was to run the plant as efficiently as possible, avoid waste of raw materials, and make a profit by shipping bottles. Two users controlled the factory, with each user able to see approximately one third of the whole plant. This form of plant was chosen because it allowed Gaver et al. to investigate how the sounds would affect the way users handled the given task and how people collaborated. It was also an opportunity to investigate how different sounds would combine to form an auditory ecology (integrated set of sounds), or soundscape. Gaver et al. related the way the different sounds in the factory combined to the way a car engine is perceived. Although the sounds are generated by multiple distinct components, these combine to form what is perceived as a unified sound. If something goes wrong, the sound of the engine will change, alerting the listener to the problem, but in addition, to a trained ear, the change in the sound would alert the listener to the nature of the problem. The sounds used to indicate the performance of the individual components of the factory were designed to reflect the semantics of the machine.

Each of the machines had a sound to indicate its status over time; for example, the bottle dispenser made the sound of clinking bottles. The rhythm of the sounds reflected the rate at which the machine was running. If a machine ran out of supplies or broke down, its sound stopped. Sounds were also added to indicate that materials were being wasted. A splashing sound indicated that liquid was being spilled; the sound of smashing bottles indicated that bottles were being lost. The system was designed so that up to 14 different sounds could be played at once. To reduce the chance that all sounds would be playing simultaneously, sounds were pulsed once a second rather than playing continuously.

An informal evaluation was undertaken where pairs of users were observed controlling the plant, either with or without sound. These observations indicated that the sounds were effective in informing the users about the state of the plant and that the users were able to differentiate the different sounds and identify the problem when something went wrong. When the sounds were used, there was much more collaboration between the two users. This was because each could hear the whole plant and therefore help if there were problems with machines that the other was controlling. In the visual only condition, users were not as efficient at diagnosing what was wrong even if they knew there was a problem.

One of the biggest advantages of auditory icons is the ability to communicate meanings which listeners can easily learn and remember, other systems (for example earcons, see the next section) use abstract sounds where the meanings are harder to learn. Problems did occur with some of the warning sounds used, as Gaver et al. (1991) indicated: "The breaking bottle sound was so compelling semantically and acoustically that partners sometimes rushed to stop the sound without understanding its underlying cause or at the expense of ignoring more serious problems" (p. 89). Another problem was that, when a machine ran out of raw materials, its sound just stopped; users sometimes missed this and did not notice that something had gone wrong.

Design Guidelines for Auditory Icons

There have been few detailed studies investigating the best ways to design auditory icons, so there is little guidance for interaction designers. Mynatt (1994) proposed the following basic design methodology: (a) Choose short sounds which have a wide bandwidth, and where length, intensity, and sound quality are roughly equal; (b) evaluate the identifiability of the auditory cues using free-form answers; (c) evaluate the learnability of the auditory cues which are not readily identified; (d) test possible conceptual mappings for the auditory cues using a repeated measures design where the independent variable is the concept that the cue will represent; (e) evaluate possible sets of auditory icons for potential problems with masking, discriminability and conflicting mappings; and (f) conduct usability experiments with interfaces using the auditory icons.

Earcons

Earcons were developed by Blattner et al. (1989). They used abstract, synthetic tones in structured combinations to create auditory messages. Blattner et al. defined earcons as "nonverbal audio messages that are used in the computer/user interface to provide information to the user about some computer object, operation or interaction" (p. 13). Unlike auditory icons, there is no intuitive link between the earcon and what it represents; the link must be learned. They use a more traditional musical approach than auditory icons.

Earcons are constructed from simple building blocks called "motifs" (Blattner et al., 1989). These short, rhythmic sequences

can be combined in different ways. Blattner et al. suggested their most important features include the following:

Rhythm: Changing the rhythm of a motif can make it sound very different. Blattner et al. (1989) described this as the most prominent characteristic of a motif.

Pitch: There are 96 different pitches in the western musical system, and these can be combined to produce a large number of different motifs.

Timbre: Motifs can be made to sound different by the use of different timbres, for example playing one motif with the sound of a violin and the other with the sound of a piano.

Register: This is the position of the motif in the musical scale. A high register means a high-pitched note, and a low register means a low-pitched note. The same motif in a different register can convey a different meaning.

Dynamics: This is the volume of the motif. It can be made to increase as the motif plays (crescendo) or decrease (decrescendo).

There are two basic ways in which earcons can be constructed. The first, and simplest, are compound earcons. These simple motifs can be concatenated to create more complex earcons. For example, a set of simple, one-element motifs might represent various system elements such as create, destroy, file, and string (see Fig. 13.4A). These could then be concatenated to form earcons (Blattner et al., 1989). In the figure, the earcon for create is a high-pitched sound that gets louder, and for destroy, it is a low-pitched sound that gets quieter. For file, there are two long notes that fall in pitch and for string, two short notes that rise. In Fig. 13.4B, the compound earcons can be seen. For the create file earcon, the create motif is simply followed by the file motif. This provides a simple and effective method for building up earcons.

Hierarchical earcons are more complex but can represent complex sound structures. Each earcon is a node in a tree and inherits properties from the earcons above it. Figure 13.5, shows a hierarchy of earcons representing a family of errors. The top level of the tree is the family rhythm. This sound just has a rhythm and no pitch; the sounds used are clicks. The rhythmic structure of level one is inherited by level two, but this time, a second motif is added where pitches are put to the rhythm. At this level, Blattner et al. (1989) suggested that the timbre should be a sine wave, which produces a colorless sound. This is done so that at level three, the timbre can be varied. At level three, the pitch is also raised by a semitone to make it easier to differentiate from the pitches inherited from level two. Other levels can be created where register and dynamics are varied.

Blattner et al. (1989) proposed the design of earcons but did not develop or test them. Brewster, Wright, and Edwards (1994) carried out a detailed evaluation of compound and hierarchical earcons based on the design proposed by Blattner et al., 1989 which involved simple system beeps and a richer design based on more complex musical timbres using psychoacoustical research (see the section on perception of sound). In these experiments, participants were presented with earcons representing families of icons, menus, and combinations of both (examples can be heard at http://www.dcs.gla.ac.uk/~stephen/demos.shtml). They heard each sound three times and then had to identify them when played back. Results showed that the more complex musical earcons were significantly more effective than both the simple beeps and Blattner's proposed design, with over 80% recalled correctly. Brewster et al. (1994) found that timbre was a much more important than previously suggested whereas pitch on its own was difficult to differentiate. The main design features of the earcons used were formalized into a set of design guidelines:

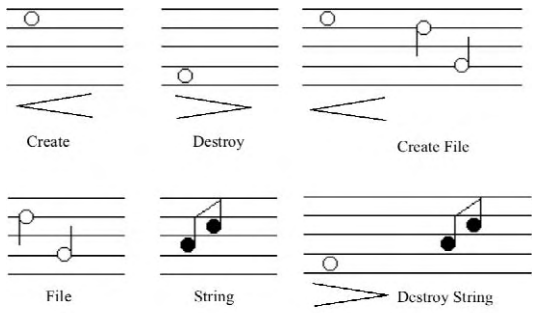


FIGURE 13.4. Compound earcons. A shows the four audio motifs 'create,' 'destroy,' 'file,' and 'string.' B shows the compound earcons 'create file' and 'destroy string' (Blattner et al., 1989).

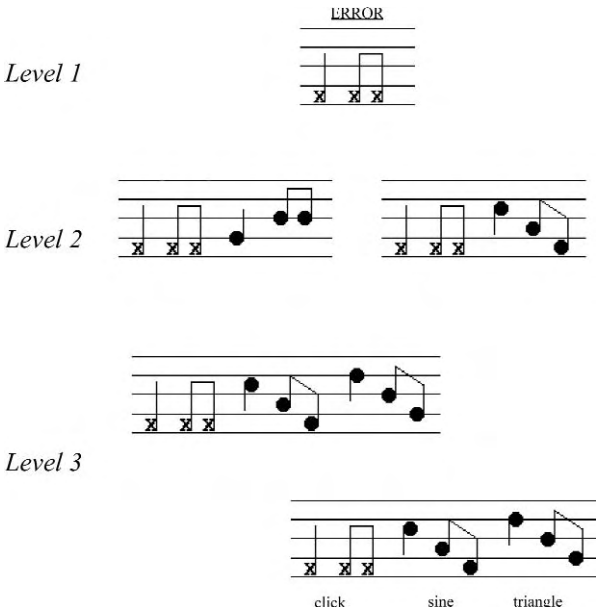


FIGURE 13.5. A hierarchy of earcons representing errors (From Blattner et al., 1989).

Timbre: This is the most important grouping factor for earcons. Use musical instrument timbres with multiple harmonics, as this helps perception and can avoid masking. These timbres are more recognizable and differentiable.

Pitch and Register: If listeners are to make absolute judgments of earcons, then pitch/register should not be used as a cue on its own. A combination of register and another parameter gives better rates of recall. If register alone must be used, then there should be large differences (two or three octaves) between earcons. Much smaller differences can be used if relative judgments are to be made. The maximum pitch used should be no higher than 5 kHz and no lower than 125 Hz–150 Hz so that the sounds are not easily masked and are within the hearing range of most listeners.

Rhythm, duration and tempo: Make rhythms as different as possible. Putting different numbers of notes in each earcon is very effective. Earcons are likely to be confused if the rhythms are similar, even if there are large spectral differences. Very short note lengths might not be noticed, so do not use very short sounds. Earcons should be kept as short as possible so they can keep up with interactions in the interface being sonified. Two earcons can be played in parallel to speed up presentation.

Intensity: This should not be used as a cue on its own because it is a major cause of annoyance. Earcons should be kept within a narrow dynamic range so that annoyance can be avoided (see section on avoiding annoyance above for more on this issue).

Major/minor mode: Lemmens (2005) showed that by changing from a major to minor key, he could change the affective responses of users to earcons. In western music, the minor mode is broadly thought of as sad and the major mode as happy. This can be used as a further cue to create differentiable earcons.

One aspect that Brewster also investigated was musical ability—as earcons are based on musical structures, is it only musicians who can use them? The results showed that more complex earcons were recalled equally well by nonmusicians as they were by musicians, indicating that they are useful to a more general audience of users.

In a further series of experiments, Brewster (1998b) looked in detail at designing hierarchical earcons to represent larger structures (with over 30 earcons at four levels). These were designed building on the guidelines above. Users were given a short training period and then were presented with sounds, and they had to indicate where the sound was in the hierarchy.

Results were again good, with participants recalling over 80% correctly, even with the larger hierarchy used. The study also looked at the learning and memorability of earcons over time. Results showed that, even with small amounts of training, users could get good recall rates, and the recall rates of the same earcons tested a week later were unchanged.

In recent work, McGookin and Brewster (2004) looked at presenting multiple earcons in parallel. This is problematic unless done carefully, as the structures used to create earcons also cause them to overlap when played in parallel. McGookin and Brewster suggested each earcon should have an onset delay and different spatial location to improve understanding. (For examples of earcons in use, see the sonically enhanced widgets and Palm III work in the section on applications of auditory output).

Comparing Auditory Icons and Earcons

Both earcons and auditory icons are effective at communicating information in sound. There is more formal evidence of this for earcons, as more basic research has looked at their design. There is less basic research into the design of auditory icons, but the systems that have used them in practice have been effective. Detailed research is needed into auditory icons to correct this problem and to provide designers with design guidance on how to create effective sounds. It may be that each has advantages over the other in certain circumstances and that a combination of both is the best. In some situations, the intuitive nature of auditory icons may make them favorable. In other situations, earcons might be best because of the powerful structure they contain, especially if there is no real-world equivalent of what the sounds are representing. Indeed, there may be some middle ground where the natural sounds of auditory icons can be manipulated to give the structure of earcons.

The advantage of auditory icons over earcons is that they are easy to learn and remember because they are based on natural sounds, and the sounds contain a semantic link to the objects they represent. This may make their association to certain, more abstract actions or objects within an interface more difficult. Problems of ambiguity can also occur when natural sounds are taken out of the natural environment and context is lost (people may have their own idiosyncratic mappings). If the meanings of auditory icons must be learned then they lose some of their advantages, and they come closer to earcons.

Earcons are abstract so their meaning must always be learned. This may be a problem, for example, in walk up and use type applications. Research has shown that little training is needed if the sounds are well designed and structured. Leplâtre and Brewster (2000) began to show that it is possible to learn the meanings implicitly while using an interface that generates the sounds as it is being used. However, some form of learning must take place. According to Blattner et al. (1989), earcons may have an advantage when there are many highly structured sounds in an interface. With auditory icons, each one must be remembered as a distinct entity because there is no structure linking them together. With earcons, there is a strong structure linking them that can easily be manipulated. There is not yet any experimental evidence to support this.

Pure auditory icons and earcons make up the two ends of a presentation continuum from representational to abstract (see Fig. 13.6). In reality, things are less clear. Objects or actions within an interface that do not have an auditory equivalent must have an abstract auditory icon made for them. The auditory icon then moves more toward the abstract end of the continuum. When hearing an earcon, the listener may hear and recognize a piano timbre, rhythm, and pitch structure as a kind of catch phrase representing an object in the interface. He or she does not hear all the separate parts of the earcon and work out the meaning from them (listeners may also try and put their own representational meanings on earcons, even if the designer did not intend it as found by Brewster, 1998b). The earcon then moves more toward the representational side of the continuum. Therefore, earcons and icons are closer than they might appear.

There are not yet any systems that use both types of sounds fully, and this would be an interesting area to investigate. Some

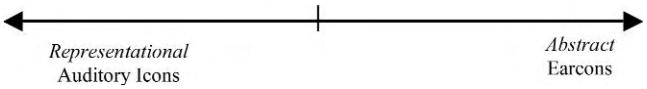


FIGURE 13.6. The presentation continuum of auditory icons and earcons.

parts of a system may have natural analogues in sound, and therefore, auditory icons could be used; other parts might be more abstract or structured and earcons would be better. The combination of the two would be the most beneficial. This is an area ripe for further research.

THE APPLICATIONS OF AUDITORY OUTPUT

Auditory output has been used in a wide range of different situations and applications. This section will outline some of the main areas of use and will highlight some of the key papers in each area (for more uses of sound see the ICAD (http://www.icad.org) or ACM CHI (http://www.acm.org/sigchi) series of conferences).

Sonic Enhancement of GUIs

One long-running strand of research in the area of auditory output is in the addition of sound to standard graphical displays to improve usability. One reason for doing this is that users can become overloaded with visual information on large, high-resolution displays. In highly complex graphical displays, users must concentrate on one part of the display to perceive the visual feedback, so that feedback from another part may be missed. This becomes very important in situations where users must notice and deal with large amounts of dynamic data. For example, imagine you are working on your computer writing a report and are monitoring several ongoing tasks, such as compiling, printing, and downloading files over the Internet. The word-processing task will take up your visual attention because you must concentrate on what you are writing. In order to check when your printout is done, the compilation has finished, or the files have downloaded, you must move your visual attention away from the report and look at these other tasks. This causes the interface to intrude into the task you are trying to perform. If information about these other tasks was presented in sound, you could continue looking at the report and hear information in the background about the other tasks. To find out how the file download was progressing, you could just listen to the download sound without moving your visual attention from the writing task.

One of the earliest pieces of work on sonic enhancement of an interface was Gaver's (1989) SonicFinder described above. This used auditory icons to present information about the Macintosh interface redundantly with the graphical display.

Brewster (1998a) investigated the addition of sound to enhance graphical buttons. An analysis of the way buttons are used was undertaken, highlighting some usability problems. It was found that the existing, visual feedback did not indicate when mispresses of a button might have occurred. For example, the selection of a graphical button is shown in Fig. 13.7 (starting with 1.A and 2.A). The button highlights when it is pressed down (Fig. 13.7 1.B and 2.B). There is no difference in feedback between a correct selection (Fig. 13.7 1.C) and a misselection (Fig. 13.7 2.C), where the user moves the mouse off the graphical button before the selection is complete. The user could therefore slip off the button, fail to press it, and get no feedback. This error can happen when the user is moving away from the button and on to some other task. For example, the user moves to a toolbar to press the "Bold" button and then moves back to the text to position the cursor to start typing. The button press and the mouse move overlap, and the button is not pressed. It is hard for the user to notice this because no feedback is given.

The problems could not easily be solved by adding more graphical feedback: the user is no longer looking at the button's location so any feedback given there will be missed. Feedback could be given at the mouse location but we cannot be sure the user will be looking there either. Brewster (1998a) designed a new button that used auditory feedback to indicate more about the state of the button. This was advantageous, as sound is omnidirectional, and the user does not need to focus attention on any part of the screen to perceive it.

Three earcons were used to improve the effectiveness of graphical buttons. An organ timbre was used for all of the sounds. When the user moved over a button, a continuous tone was played at 130 Hz at a volume just above the background sound level. This informed the user the cursor was over the target (but could easily be habituated). When the mouse was pressed down over the graphical button, a continuous tone was played at 261 Hz. The third sound indicated that the graphical button had been successfully selected. This sound consisted of two short tones with a pitch of 1046 Hz and duration of 40ms. This sound was not played if a slip-off error occurred. If the user pressed the button very quickly, then only the success sound was played to avoid unnecessary feedback.

An experimental evaluation of these sounds was undertaken. Results showed that users recovered from slip-off errors significantly faster and with significantly fewer mouse clicks when sounds were present in the buttons. Users also significantly preferred the buttons with sound when asked to rate subjective preference. They also did not rate the buttons as more annoying than the standard graphical ones. An interesting point to note was that the use of no sound could be attention grabbing when a sound was expected. The participants could easily recognize a slip off due to the demanding nature of the success sound not being played. This is important as reducing the amount of feedback presented is one way to make sure that it is not annoying.

Many other widgets have been successfully sonified. Beaudouin-Lafon and Conversey (1996) showed that nonspeech sounds could improve usability of scrollbars, Maury, Athenes, and Chatty (1999) and Marila (2002) added sounds to improve menu selections in drop-down menus. Ronkainen and Pasanen (2005) did several studies into the design of audio feedback for buttons. Brewster (1998a) investigated a wide range of different widgets including scroll-bars, menus, progress bars, tool palettes, and drag and drop. These widgets have been included in a toolkit (Crease, Gray, & Brewster, 2000) that designers can use to add sound easily to their interfaces.

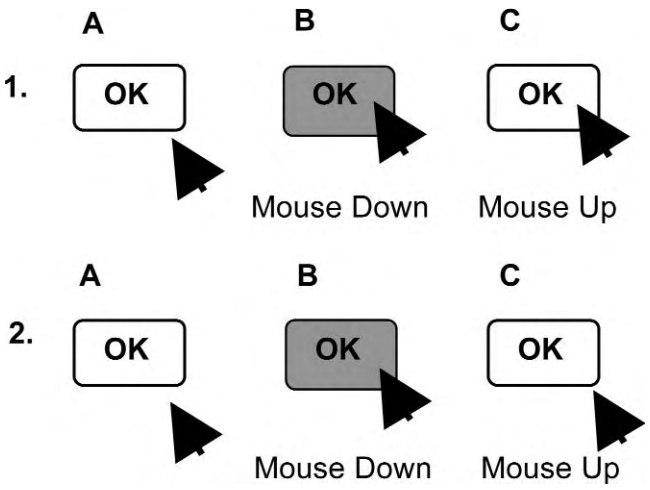


FIGURE 13.7. The visual feedback presented by a graphical button when selected. 1 shows a correct selection and 2 shows a slip-off (From Brewster, 1998a).

Sound for Users with Visual Impairments

One of the most important uses for nonspeech sound is in interfaces for people with visual disabilities. One of the main deprivations caused by blindness is the problem of access to information. A blind person will typically use a screen reader and a voice synthesizer to use a computer. The screen reader extracts textual information from the computer's video memory and sends it to the speech synthesizer to speak it. This works well for text but not well for the graphical components of current user interfaces. It is still surprising to find that many commercial applications used by blind people make little use of nonspeech sound, concentrating on synthetic speech output. This is limiting (as discussed above), as speech is slow, it can overload short-term memory, and it is not good for presenting certain types of information; for example, it is not possible to render many types of images via speech, so these can become inaccessible to blind people. One reason for sound's lack of use has been how to employ it effectively. As Edwards (1995) said, "Currently the greatest obstacle to the exploitation of the variety of communications channels now available is our lack of understanding of how to use them" (p. ii). The combination of speech and nonspeech sounds can increase the amount of information presented to the user. As long as this is done in a way that does not overload the user, then it can improve access to information. Some of the main research into the use of nonspeech auditory in interfaces for blind people will now be described (for more on perceptual impairments in general see chapter 43).

Soundtrack was an early attempt to create a word processor designed to be used by blind persons and was developed by Edwards (1989). It used earcons and synthetic speech as output, and it was designed so that the objects a sighted user would see in an interface, for example menus and dialogues, were replaced by auditory equivalents that were analogies of their visual counterparts. Its interface was constructed from auditory objects with which the user could interact. They were defined by a location, a name, a sound, and an action. They were arranged into a grid of two layers, analogous to menus (see Fig. 13.8).

Each auditory object made a sound when the cursor entered it, and these could be used to navigate rapidly around the screen. Soundtrack used sine waves for its audio feedback. Chords were built up for each menu, dependent on the number of menu items. For the edit menu, a chord of four notes was played because there were four menu items within it (cut, copy, paste, and find).

The base sounds increased in pitch from left to right—as in the normal representation of a musical scale (for example on a piano) and the top layer used higher pitches than the bottom. Using these two pieces of information a user could quickly find his or her position on the screen. If any edge of the screen was reached, a warning sound was played. If at any point the user got lost or needed more precise information, he or she could click on an object and it would speak its name.

The approach taken in Soundtrack was to take the visual interface to a word processor and translate it into an equivalent auditory form. The Mercator system (Mynatt & Edwards, 1995; Mynatt & Weber, 1994) took a broader approach. The designers' goal was to model and translate the graphical interfaces of X Windows applications into sound without modifying the applications (and thus create a more general solution than Soundtrack's). Their main motivation was to simulate many of the features of graphical interfaces to make graphical applications accessible to blind users and keep coherence between the audio and visual interfaces so that blind and sighted users could interact and work together on the same applications. This meant that the auditory version of the interface had to facilitate the same mental model as the visual one. This did not mean that they translated every pixel on the screen into an auditory form; instead, they modeled the interaction objects that were present. Modeling the pixels exactly in sound was ineffective due to the very different nature of visual and auditory media and the fact that graphical interfaces had been optimized to work with the visual sense (for example, the authors claim that an audio equivalent of overlapping windows was not needed as overlapping was just an artifact of a small visual display). Nonspeech sound was an important aspect of their design to make the iconic parts of a graphical interface usable.

Mercator used three levels of nonspeech auditory cues to convey symbolic information presented as icons in the visual interface. The first level addressed the question, what is this ob-

File Menu	Edit Menu	Sound Menu	Format Menu
Alert	Dialog	Document 1	Document 2

FIGURE 13.8. The Soundtrack's main screen (From Edwards, 1989).

ject? In Mercator, the type of an interface object was conveyed with an auditory icon. For example, touching a window sounded like tapping a piece of glass, container objects sounded like a wooden box with a creaky hinge, and text fields used the sound of a manual typewriter. While the mapping was easy for interface components such as trashcan icons, it was less straightforward components that did not have simple referents in reality (e.g., menus or dialogue boxes, as discussed in the section on sonic enhancement of GUIs). In Mercator, auditory icons were also parameterized to convey detailed information about specific attributes, such as menu length. Global attributes were also mapped into changes in the auditory icons. For example, highlighting and graying out are common to a wide range of different widgets. To represent these, Mynatt and Edwards (1995) used sound filters. A low-pass filter was used to make the sound of a grayed out object duller and more muffled.

Sonification

Building on the work about accessibility emerges the idea of making data accessible. Sonification, or visualization in sound, can be used to present complex data nonvisually. There are many situations where sonification can also be useful for sighted users (if they only have access to a small screen for example), or in combination with graphical feedback in multimodal visualization systems. Sonification is defined as "the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation" (Kramer & Walker, 1999). The range of sonification goes from the clicks of the Geiger counter to multidimensional information presentation of stock market data.

Mansur et al. (1985) performed one of the most significant studies presenting data in sound. Their study, which laid out the research agenda for subsequent research in sound graphs, used sound patterns to represent two-dimensional line graphs. The value on the *y*-axis of the graph was mapped to pitch and the *x*-axis to time; this meant that a listener could hear the graph rise and fall over time in a similar way that a sighted person could see the line rising and falling. This is the basic technique used in most sonification systems.

They found that their approach was successful in allowing distinctions to be made between straight and exponential graphs, varying monotonicity in graphs, convergence, and symmetry. However, they did find that there were difficulties in identifying secondary aspects of graphs, such as the slope of the curves. They suggested that a full sound graph system should contain information for secondary aspects of the graph, such as the first derivative. Their suggestion was to encode this information by adding more overtones to the sound to change the timbre. They also suggested utilizing special signal tones to indicate a graph's maxima or minima, inflection points, or discontinuities. Many other studies have been undertaken to develop this presentation technique further (Flowers & Hauer, 1992, 1995; Walker, 2002).

Walker and Cothran (2003) produced the Sonification Sandbox to allow designers to design sound graphs easily and rapidly. The software allows "users to independently map several data sets to timbre, pitch, volume, and pan, with full control over the default, minimum, maximum, and polarity for each attribute." This gives auditory interface designers the chance to prototype sonifications, without having to create their own, custom-made applications. (The software is freely available from http://sonify.psych.gatech.edu/research.)

Sound for Mobile and Ubiquitous Computing

One of the major growth areas in computing at the beginning of the 21st century has been in mobile and ubiquitous computing. People no longer use computers just while at a desk. Mobile telephones, personal digital assistants (PDAs), and handheld computers are now widely used (see chapter 23 for more on mobile and ubiquitous computing). One problem with these devices is that there is a very limited amount of screen space on which to display information: the screens are small as the devices must be able to fit into the hand or pocket to be easily carried. Small screens can easily become cluttered with information and widgets and this presents a difficult challenge for interface designers.

The graphical techniques for designing interfaces on desktop interfaces do not apply well to handheld devices. Screen resources are limited; memory and processing power are much reduced from desktop systems. However, in many cases, interface designs and interaction techniques have been taken straight from standard desktop graphical interfaces (where screen space and other resources are not a problem) and applied directly to mobile devices. This has resulted in devices that are hard to use, with small text that is hard to read, cramped graphics and little contextual information. Speech and nonspeech sounds an important way of solving these problems.

Another reason for using sound is that if users are performing tasks while walking or driving, they cannot devote all of their visual attention to the mobile device. Visual attention must remain with the main task for safety. It is therefore hard to design a visual interface that can work well under these circumstances. An alternative, sonically enhanced interface would require less visual attention, and therefore potentially interfere less in the main activity in which the user is engaged.

Three main pieces of work are surveyed in this section, covering the main approaches taken in this area. The first adds sound to the existing interface of a mobile computer to improve usability, the second creates a purely auditory interface for a mobile, and the third creates an ambient auditory environment.

Brewster (2002) developed the ideas of sonified buttons described in the section on sonic enhancement of GUIs and applied them to buttons on the 3Com Palm series of pen-based handheld computers. Many of the same feedback problems with buttons apply in handhelds as in desktops, but are worse as the screen is smaller (and may be hard to see when the device is moving or the sun is shining). In addition, there is the problem of the stylus (or finger) obscuring the target on the display, which makes it difficult for users to know when they are pressing in the correct place. Simple earcons were used to overcome the problems. One aim of the work was to see the effects when users were on the move and to see if adding audio could reduce the size of the widgets so that screen space could be saved.

In general, the results confirmed those of the previous study. Subjective workload in the sonically enhanced buttons was re-

duced, as compared to their silent counterparts. The addition of sound allowed the participants to enter significantly more five-digit strings than in the corresponding silent treatment, with smaller sonic buttons as effective as larger silent ones. When walking, there was a 20% drop in performance overall, with the sonic interface still performing better than the standard one. Participants walked further when sound was added, and small buttons with sound allowed as much text to be entered as the large, silent buttons. The suggested reason for this was that users did not have to concentrate so much of their visual attention on the device; much of the feedback needed was in sound, so they could look where they were going. This would therefore allow the size of items on the display to be reduced without a corresponding drop in usability.

Sawhney and Schmandt (1999, 2000) developed a wearable computer-based personal messaging audio system called "nomadic radio" to deliver information and messages to users on the move. One of the aims of this system was to reduce the interruptions to a user caused by messages being delivered at the wrong time (for example mobile telephone calls being received in a meeting, or a PDA beeping to indicate an appointment in the middle of a conversation). In the system, users wore a microphone and shoulder-mounted loudspeakers that provide a basic planar 3D audio environment (see section on three dimensional sound) through which the audio was presented. A clock face metaphor was used with 12:00 in front of the user's nose, 3:00 by the right ear, 6:00 directly behind the head, and so forth. Messages were then presented in the position appropriate to the time that they arrived. The advantage of the 3D audio presentation (as described above) is that it allows users to listen to multiple, simultaneous sound streams and to distinguish and separate each one (the cocktail party effect shows that listeners can attend one stream of sound amongst many, but also monitor the others in case they need attention; Arons, 1992).

The system used a context-based notification strategy that dynamically selected the appropriate notification method based on the user's focus of attention. Seven levels of auditory presentation were used from silent to full speech rendering. If the user was engaged in a task, then the system was silent and no notification of an incoming call or message would be given (so as not to cause an interruption). The next level used ambient cues (based on auditory icons) with sounds like running water indicating that the system was operational. These cues were designed to be easily habituated but to let the user know that the system was working. The next level was a more detailed form of auditory cue, giving information on system events, task completions, and mode transitions. For example, a ringing telephone sound was used to indicate the arrival of voicemail. These were more attention grabbing cues than the ambient cues and would only be played if the user was not fully occupied. The next four levels of cue used speech, expanding from a simple message summary up to the full text of a voicemail message. These might be used if the person wearing nomadic radio was not involved in tasks that required detailed attention. The system attempted to work out the appropriate level to deliver the notifications by listening to the background audio level near the user (using the built-in microphone) and by determining whether the user was speaking. For example, if the user was speaking, the system might use an ambient cue so as not to interrupt the conversation. Users could also press a button on the device to indicate they were busy and turn it to silent.

Three-dimensional sound has been combined with gestures to create interactions where users can point at sound sources to choose them. An early example was from Cohen (1993) who created audio windows that users could manipulate with gestures, much as windows on a desktop computer could be controlled. Brewster, Lumsden, Bell, Hall, and Tasker (2003) made this idea mobile and created a soundscape of audio sources around a listener's head that presented different types of information. Users nodded at a sound source of interest to select it. A simple study showed that users could walk and nod to select items, but that there were many issues with sound placement and feedback. Further study by Marentakis and Brewster (2004) looked at different types of gestures and feedback to improve the quality of mobile 3D audio interactions.

There has been much work in the area of notification systems using audio for ambient displays. Carefully designed nonspeech audio can grab attention and then fade into the background. An early example was Audio Aura by Mynatt, Back, Want, Baer, and Ellis (1998), which aimed "to provide serendipitous information, via background auditory cues, that is tied to people's physical actions in the workplace" (p. 566). In a similar way to nomadic radio, Audio Aura used auditory icons to provide background information that did not distract users.

The system used active badges so that the location of users could be identified and appropriate audio cues given, along with wireless headphones so that users could hear the sounds without distracting others. The location information from the active badges was combined with other data sources such as online calendars and e-mail. Changes in this information triggered audio cues sent to the user through the headphones.

Here are some examples of how the system might be used. In the first the user might go the office coffee room and as he or she enters the room hear information about the number and type of e-mail messages currently waiting. This would give the user a cue as to whether to stay and talk to colleagues or go back to the office to answer the messages. In the second example, a user goes to a colleague's office but the occupant is not there. Audio Aura would play sounds indicating if the occupant has been in recently or away for a longer period. The authors were keen to make sure the sounds were not distracting and attention grabbing—they were meant to give background information and not to be alarms. To this end, great care was taken with the cue design. They attempted to design sonic ecologies—groups of sounds that fitted together into a coherent whole. For example, one set of cues was based on a beach scene. The amount of new e-mail was mapped to seagull cries: the more mail the more the gulls cried. Group activity levels were mapped to the sound of surf: the more activity going on within the group the more active the waves became. These cues were very subtle and did not grab users' attentions, but some learning of the sounds would be needed, as they are quite abstract.

This work was taken on and implemented in a realistic environment by Kilander and Lönnqvist (2001). They created a Weakly Intrusive Ambient Soundscape (WISP), where states in the computational or physical ubiquitous computing environment are presented as subtle, nonintrusive sound cues based on auditory icons, with each cue "sufficiently nonintrusive to be accepted without disturbing the focus of the task at hand, while distinctive enough to be separable from other cues" (Kilander

& Lonnqvist, 2002, p. 1). They describe a meeting room scenario where devices such as handheld computers, public PCs, and clocks might all be able to make sounds and give cues for ambient awareness. The level of intrusiveness of the sounds could be varied. For low intrusiveness, a quiet sound with lots of reverb was played, making the cue sound far away and almost inaudible; for high intrusiveness, sharp sounds with no reverb were played.

One problem with the system was choice of sounds; users could be detected by the environment, their personal sound mappings, and parameters chosen. However, these mappings could conflict with others' choices; for example, two users might use the same sound cue for different events. This could be solved to some extent with local presentation of the sounds, as in nomadic radio, but global cues would be more of a problem for these types of systems.

CONCLUSIONS

Research into the use of nonspeech sounds for information display at the user interface began in the early 1990s, and there has been rapid growth since then. It has shown its benefits in a wide range of different applications from systems for blind people to ubiquitous computing. There are many good examples that designers can look at to see how sounds may be used effectively, and design guidelines are now starting to appear.

Two areas are likely to be important in its future growth. The first is in combining sound with other senses (vision, tactile, force-feedback, etc.) to create multimodal displays that make the most of the all the senses available to users. This is an area ripe for further investigation, and many interesting interaction problems can be tackled when multiple senses are used together. Key questions here are around what sound is best for and how to combine it with the others most effectively. The second area in which nonspeech sound has a large part to play is with mobile and wearable computing devices (again also in a multimodal form). Small screens cause many difficult presentation problems, and this is exactly the situation where sound has many advantages—it does not take up any precious screen space, and users can hear it even if they cannot look at their device. In a ubiquitous setting, there may not even be a screen at all, and sound can provide information on the services available in a particular environment in a nonintrusive way.

References

Alty, J. L. (1995). *Can we use music in human-computer interaction?* Paper presented at British Computer Society Human conference (HCI'95), Huddersfield, UK.

Arons, B. (1992, July). A review of the cocktail party effect. *Journal of the American Voice I/O Society, 12*, 35–50.

Baddeley, A. (1997). *Human memory: Theory and practice*. London: Psychology Press Ltd.

Bagwell, C. (1998). *Audio file formats FAQ*. Retrieved November, 2005, from http://www.cnpbagwell.com/audio.html

Barker, P. G., & Manji, K. A. (1989). Pictorial dialogue methods. *International Journal of Man-Machine Studies, 31*, 323–347.

Beaudouin-Lafon, M., & Conversy, S. (1996). *Auditory illusions for audio feedback.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI'96), Vancouver, Canada.

Berglund, B., Harder, K., & Preis, A. (1994). Annoyance perception of sound and information extraction. *Journal of the Acoustical Society of America, 95*(3), 1501–1509.

Blattner, M., & Dannenberg, R. B. (Eds.). (1992). *Multimedia interface design*. New York: ACM Press, Addison-Wesley.

Blattner, M., Sumikawa, D., & Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction, 4*(1), 11–44.

Blauert, J. (1997). *Spatial hearing*. Cambridge, MA: MIT Press.

Bly, S. (1982). *Sound and computer information presentation.* Unpublished PhD Thesis No. UCRL53282, Lawrence Livermore National Laboratory, Livermore, CA.

Brewster, S. A. (1998a). The design of sonically-enhanced widgets. *Interacting with Computers, 11*(2), 211–235.

Brewster, S. A. (1998b). Using nonspeech sounds to provide navigation cues. *ACM Transactions on Computer-Human Interaction, 5*(3), 224–259.

Brewster, S. A. (2002). Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing, 6*(3), 188–205.

Brewster, S. A., Lumsden, J., Bell, M., Hall, M., & Tasker, S. (2003). *Multi-modal 'eyes-free' interaction techniques for wearable devices.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI 2003), Fort Lauderdale, FL.

Brewster, S. A., Wright, P. C., & Edwards, A. D. N. (1994). A detailed investigation into the effectiveness of earcons. In G. Kramer (Ed.), *Auditory display* (pp. 471–498). Reading, MA: Addison-Wesley.

Buxton, W. (1989). Introduction to this special issue on nonspeech audio. *Human Computer Interaction, 4*(1), 1–9.

Buxton, W., Gaver, W., & Bly, S. (1991). *Tutorial number 8: The use of nonspeech audio at the interface.* Paper presented at Association for Computing Machinery Computer Human Interaction conference CHI'91, New Orleans, LA.

Chowning, J. (1975). Synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society, 21*(7), 526–534.

Cohen, M. (1993). Throwing, pitching and catching sound: Audio windowing models and modes. *International Journal of Man-Machine Studies, 39*, 269–304.

Crease, M. C., Gray, P. D., & Brewster, S. A. (2000). *Caring, sharing widgets.* Paper presented at British Computer Society Human (HCI 2000) conference, Sunderland, UK.

Edwards, A. D. N. (1989). Soundtrack: An auditory interface for blind users. *Human Computer Interaction, 4*(1), 45–66.

Edwards, A. D. N. (Ed.). (1995). *Extra-ordinary human-computer interaction*. Cambridge, UK: Cambridge University Press.

Edworthy, J., Loxley, S., & Dennis, I. (1991). Improving auditory warning design: Relationships between warning sound parameters and perceived urgency. *Human Factors, 33*(2), 205–231.

Flowers, J. H., & Hauer, T. A. (1992). The ear's versus the eye's potential to assess characteristics of numeric data: Are we too visuocentric? *Behavior Research Methods, Instruments, and Computers, 24*, 258–264.

Flowers, J. H., & Hauer, T. A. (1995). Musical versus visual graphs: Cross-modal equivalence in perception of time series data. *Human Factors, 37*, 553–569.

Frysinger, S. P. (1990). *Applied research in auditory data representation.* Extracting meaning from complex data: Processing, display, interaction. Proceedings of the SPIE/SPSE symposium on electronic imaging, Springfield, VA.

Gaver, W. (1989). The sonicfinder: An interface that uses auditory icons. *Human Computer Interaction, 4*(1), 67–94.

Gaver, W. (1997). Auditory interfaces. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 1003–1042). Amsterdam: Elsevier.

Gaver, W., Smith, R., & O'Shea, T. (1991). *Effective sounds in complex systems: The ARKola simulation.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI'91), New Orleans, LA.

Gelfand, S. A. (1981). *Hearing: An introduction to psychological and physiological acoustics*. New York: Marcel Dekker Inc.

Kilander, F., & Lonnqvist, P. (2001). A weakly intrusive ambient soundscape for intuitive state perception. In G. Doherty, M. Massink, & M. Wilson (Eds.), *Continuity in future computing systems* (pp. 70–74). Oxford, UK: CLRC.

Kilander, F., & Lonnqvist, P. (2002). *A whisper in the woods—an ambient soundscape for peripheral awareness of remote processes.* Paper presented at International Conference on Auditory Display (ICAD 2002), Kyoto, Japan.

Kramer, G. (1994a). An introduction to auditory display. In G. Kramer (Ed.), *Auditory display* (pp. 1–77). Reading, MA: Addison-Wesley.

Kramer, G. (Ed.). (1994b). *Auditory display* (Vol. Proceedings volume XVIII). Reading, MA: Addison-Wesley.

Kramer, G., & Walker, B. (Eds.). (1999). *Sonification report: Status of the field and research agenda*. Santa Fe: The International Community for Auditory Display.

Lemmens, P. (2005). *Using the major and minor mode to create affectively-charged earcons.* Paper presented at International Conference on Auditory Display (ICAD 2005), Limerick, Ireland.

Leplâtre, G., & Brewster, S. A. (2000). *Designing nonspeech sounds to support navigation in mobile phone menus.* Paper presented at International Conference on Auditory Display (ICAD 2000), Atlanta, GA.

Mansur, D. L., Blattner, M., & Joy, K. (1985). Sound-graphs: A numerical data analysis method for the blind. *Journal of Medical Systems, 9*, 163–174.

Marentakis, G., & Brewster, S. A. (2004). *A study on gestural interaction with a 3D audio display.* Paper presented at MobileHCI 2004, Glasgow, UK.

Marila, J. (2002). *Experimental comparison of complex and simple sounds in menu and hierarchy sonification.* Paper presented at International Conference on Auditory Display (ICAD 2003), Kyoto, Japan.

Maury, S., Athenes, S., & Chatty, S. (1999). *Rhythmic menus: Toward interaction based on rhythm.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI'99), Pittsburgh, PA.

McCormick, E. J., & Sanders, M. S. (1982). *Human factors in engineering and design* (5th ed.). New York: McGraw-Hill.

McGookin, D. K., & Brewster, S. A. (2004). Understanding concurrent earcons: Applying auditory scene analysis principles to concurrent earcon recognition. *ACM Transactions on Applied Perception, 1*(2), 120–155.

Miranda, E. R. (1998). *Computer sound synthesis for the electronic musician*. Oxford, UK: Focal Press.

Moore, B. C. (2003). *An introduction to the psychology of hearing* (5th ed.). Oxford, UK: Elsevier Science.

Mynatt, E. D. (1994). *Designing with auditory icons: How well do we identify auditory cues?* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI '94), Boston, MA.

Mynatt, E. D., Back, M., Want, R., Baer, M., & Ellis, J. B. (1998). *Designing audio aura.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (ACM CHI'98), Los Angeles, CA.

Mynatt, E. D., & Edwards, K. (1995). Chapter 10: Metaphors for nonvisual computing. In A. D. N. Edwards (Ed.), *Extra-ordinary human-computer interaction* (pp. 201–220). Cambridge, UK: Cambridge University Press.

Mynatt, E. D., & Weber, G. (1994). *Nonvisual presentation of graphical user interfaces: Contrasting two approaches.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI'94), Boston, MA.

Neuhoff, J. G. (Ed.). (2004). *Ecological psychoacoustics*. San Diego, CA: Elsevier Academic Press.

Patterson, R. D. (1982). *Guidelines for auditory warning systems on civil aircraft* (CAA Paper No. 82017), London, Civil Aviation Authority.

Patterson, R. D. (1989). Guidelines for the design of auditory warning sounds. *Proceeding of the Institute of Acoustics, Spring Conference, 11*(5), 17–24.

Perrott, D., Sadralobadi, T., Saberi, K., & Strybel, T. (1991). Aurally aided visual search in the central visual field: Effects of visual load and visual enhancement of the target. *Human Factors, 33*(4), 389–400.

Pohlmann, K. (2005). *Principles of digital audio* (5th ed.). New York: McGraw-Hill.

Roads, C. (1996). *The computer music tutorial*. Cambridge, MA: MIT Press.

Ronkainen, S., & Pasanen, L. (2005). *Effect of aesthetics on audio-enhanced graphical buttons.* Paper presented at International Conference on Auditory Display (ICAD 2005), Limerick, Ireland.

Sawhney, N., & Schmandt, C. (1999). *Nomadic radio: Scalable and contextual notification for wearable messaging.* Paper presented at Association for Computing Machinery Computer Human Interaction conference (CHI'99), Pittsburgh, PA.

Sawhney, N., & Schmandt, C. (2000). Nomadic radio: Speech and audio interaction for contextual messaging in nomadic environments. *ACM Transactions on Human-Computer Interaction, 7*(3), 353–383.

Strybel, T., Manligas, C., & Perrott, D. (1992). Minimum audible movement angle as a function of the azimuth and elevation of the source. *Human Factors, 34*(3), 267–275.

Thimbleby, H. (1990). *User interface design*. New York: ACM Press, Addison-Wesley.

Walker, B. N. (2002). Magnitude estimation of conceptual data dimensions for use in sonification. *Journal of Experimental Psychology, Applied, 8*, 211–221.

Walker, B. N., & Cothran, J. (2003). *Sonification sandbox: A graphical toolkit for auditory graphs.* Paper presented at International Conference on Auditory Display (ICAD 2003), Boston, MA.

•14•

NETWORK-BASED INTERACTION

Alan Dix
Lancaster University

Introduction **266**
Structure 266
About Networks **266**
Networks as Enablers **267**
Things that are Only Possible with Networks 267
Remote Resources 267
People 267
Physical Things 267
Data 268
Computation 268
Applications 269
Networks as Mediators **270**
Issues and Problems Because of Networks 270
Network Properties 270
Bandwidth and Compression 270
Latency and Start-Up 271
Jitter and Buffering 271
Reliability and Loss, Datagram, and Connection-Based Services 273
Quality of Service (QoS) and Reservation 273
Encryption, Authentication, and Digital Signatures 273
UI Properties 274
Network Transparency 274
Delays and Time 274
Coping Strategies 275
Timeliness of Feedback/Feedthrough, Pace 275
Race Conditions and Inconsistent Interface States 276
Awareness 276
Media Issues 277
Interactive Conversation and Action 277
Reliability 278
Sound and Vision 278
Compression 278
Jitter 278
Broadcast and Prerecorded Media 278
Public Perception: Ownership, Privacy, and Trust 279
Networks as Subjects **279**
Understanding and Managing Networks 279
Network Models 280
Layers 280
Protocols 281
Internetworking and Tunneling 281
Routing 282
Addresses 282
All Together 283
Decentralizing: Peer–Peer and Ad-Hoc Networks 283
Network Management 283
Network Awareness 284
Network Confusion 285
Exploiting the Limitations: Seamfulness and Virtual Locality 285
The Network Within 285
Networks as Platforms **286**
Algorithms and Architectures for Distributed Interfaces 286
Accessing Shared Objects 286
Locking 287
Replication 287
Optimistic Concurrency for Synchronous Editing 287

Groupware Undo 288
Real Solutions? 288
Architectures for Networked Systems 289
Supporting Infrastructure 289
Awareness Servers 289
Notification Servers 289
Event/Messaging Systems 289
Resource Discovery 289
Web Service and Remote Procedure Call 289
History, Futures, and Paradigm Shifts 290
History 290
Paradigm Shift 291
Futures (Near) 291
References 292

INTRODUCTION

In some ways this chapter could be seen as redundant in an HCI book—surely networks are just an implementation mechanism, a detail below the surface; all that matters are the interfaces that are built on them. On the other hand, networked interfaces, especially the web—and, increasingly, mobile devices—have changed the way we view the world and society. Even those bastions of conservatism, the financial institutions, have found themselves in sea change, and a complete re-structuring of the fundamentals of businesses is just an implementation detail.

Structure

The chapter will begin with a brief overview of types of networks and then deal with network-based interaction under four main headings:

- **Networks as Enablers**—things that are only possible with networks
- **Networks as Mediators**—issues and problems because of networks
- **Networks as Subjects**—understanding and managing networks
- **Networks as Platforms**—algorithms and architectures for distributed interfaces

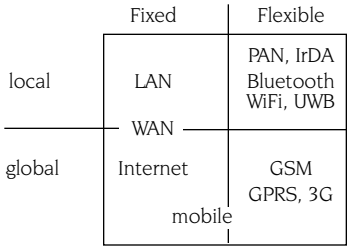
In addition, there will be a section taking a broader view of the history and future of network interaction and the societal effects and paradigm changes engendered, especially by more recent developments in global and wireless networking.

ABOUT NETWORKS

The word *network* will probably make many think of accessing the Internet and the web. Others may think of a jumble of Ethernet wires between the PCs in their office. In fact, the range of networking standards, including physical cabling (or lack of cabling) and the protocols that computers use to talk along those cables, is extensive. Although most of the wire-based networks have been around for some time, they are in a state of flux, due to increases in scale and the demands of continuous media. In the wireless world things are changing even more rapidly with two new generations of data service being introduced over the next two years.

As an aid to seeing the broader issues surrounding these changing (and, in some cases, potentially ephemeral) technologies, we can use the following two dimensions to classify them:

- **Global vs. Local**
  How spatially distant are the points connected—ranging from machines in the same room (IrDa, Bluetooth), through those in a building/site (LAN) to global networks (Internet, mobile phone networks)?
- **Fixed vs. Flexible**
  How permanent are the links between points of the network, from physically fixed machines, to self-reconfiguring devices that recognize other devices in their vicinity?



The fixed vs. flexible dimension is almost, but not quite, terrestrial vs. wireless. The "not quite" is because fixed networks increasingly involve wireless links. Also, it is often possible, when visiting another organization, to plug a portable computer into a (wired) Ethernet network and find you have access to the local printers, Internet connections, and so on—flexible wire-based networking.

Let's look at a few network technologies against these dimensions. Traditional office LANs (local area networks) are squarely in the local–fixed category, whereas the Internet is largely global–fixed. Corporate WANs (wide area networks), connecting offices within the same national or international company, sit somewhere between.

Mobile phones have been placed within the global–fixed category as well. This may seem strange; the phone can go anywhere. However, the interconnections between phones are fixed and location-independent. If two mobile phones are in the same room it is no easier to connect between them than if they were at opposite ends of the earth (bar a shorter lag time perhaps).

Similarly, the Internet, although increasingly accessible through mobile devices and phones, is largely based on fixed domain names, IP numbers, and URLs.

Given the ideas that the placement of mobile phones is a little ambiguous and it is possible to detect the location of phones and thus deliver location-based content, some of the phone technologies have been listed in the global–flexible category. These include GSM (Global System for Mobile Communications), GPRS (General Packet Radio Service), and 3G. There is obviously a steadily increasing data rate, and third-generation

services are able to cope with heavy media content including live video; in fact, the "killer app" is live sports highlights! However, the most significant differences are the charging and connectivity model. With GSM you connect when required to the Internet and this is treated like any other telephone call, usually meaning pay-per-minute whilst connected. In contrast, second- and third-generation services are based on sending small packets of data (the P in the GPRS acronym). The connection to the Internet is treated as "always on" and packets of data are sent to or from the phone as required. Charging is also typically by data use or by fixed charge.

In the local–flexible category there is a host of existing and emerging technologies. At the most mundane are the now ubiquitous WiFi networks and hotspots (based on the 802.11 protocol) (IEEE, 2001). These merely treat the machine the same as if it were plugged into the local fixed network. At a more local scale, infrared (IrDa) enabled devices can talk to one another if their infrared sensors are within line-of-sight, and Bluetooth (Bluetooth, 2001) or emerging wireless technologies such as ZigBee or UWB (Ultra Wide Band) (Zigbee, 2006; WiMedia, 2006) allow flexible connections between personal devices. With these, a laptop can use a mobile-phone modem, or a Bluetooth hands-free headset can connect to a phone without having to plug in with a piece of wire.

These same technologies can also be used to establish local connections with printers or other devices or even track people using the unique addresses that are often broadcast continually. Thus they offer both the opportunities of accessing fixed public equipment through personal devices, as well as the threat of surveillance and hacking everywhere!

Finally, research in wearable computers has suggested using the body itself as the connection between worn devices in a personal area network (PAN) (Zimmerman, 1996). The future is networked and we will become the network.

On the whole we have seen in the last 10 years the main focus of network-based interaction has moved anti-clockwise in this picture from fixed/local networks (mainly LAN), through fixed global networks (the Internet and web explosion), through global mobile networks (mostly phone-based, but including WAP (Wireless Application Protocol), i-mode, etc.), and moving towards flexible local connections between devices. In both the local and global spaces there has also been a growth of less centrally controlled networking with peer–peer services establishing decentralized applications over the Internet, and wireless ad-hoc networks allowing machines to establish networks with no fixed infrastructure.

NETWORKS AS ENABLERS

Things That Are Only Possible With Networks

It can be the case that the network is no more than an implementation detail; for example, using a networked disk rather than a local one. However, there are also many applications, like videoconferencing, which are only possible because the network is there. The key feature of networks is the access to remote resources of some kind or other.

Remote Resources

Four kinds of remote things are made accessible by networks:

- People
- Physical things
- Data
- Computation

These things may be remote because they are far away from where you normally are, or because you are yourself on the move and hence away from your own resources (colleagues, databases, etc.). Hence, mobility can create a need for any or all the above.

People

Networks mean we can communicate and work with others in distant places. This is often a direct action, such as e-mailing someone or engaging in a videoconference. These are all the normal study of CSCW (Computer-Supported Cooperative Work) and groupware (see chapter 29).

Interaction with remote people may also be indirect. Recommender systems gather information about people's preferences and use this to suggest further information, services, or goods based on their own preferences and those of others who have similar tastes (Resnick & Varian, 1997). Because the people making recommendations are in different locations, the data on who selected what must be stored centrally. If you have been suggested books at Amazon.com, you have experienced a recommender system.

Collaborative virtual environments also offer the ability for remote people to interact, but by embedding them within an apparently local virtual-reality world. Although the person you are dealing with may be half a world away, her avatar (a virtual presence, perhaps a cartoon character, photo, or robot-like creature) may seem only a few yards or meters away in the virtual world.

Physical Things

We can also view and control remote things at a distance. For example, live web cams in public places allow us to see things (and people) there. Similarly, the cameras mounted around rockets as they prepare to take off (and are then usually destroyed during the launch) allow the mission controllers to monitor critical aspects of the physical system, as do the numerous telemetry sensors which will also be related via some sort of closed network. And of course the launch command itself will be relayed to the rocket by the same closed network as will the ongoing mission, perhaps the Mars robots, via wireless links.

In the rocket example it would be dangerous to be in the actual location; in other circumstances it is merely expensive or inconvenient. Telescopes are frequently mounted in distant parts of the world where skies are clearer than those above the laboratories to which they belong. In order to avoid long international trips to remote places, some of these telescopes now have some form of remote control and monitoring using the Internet (Lavery, Kilgourz, & Sykeso, 1994).

At a more personal level the systems within certain high-end cars are controlled using a within-car network (called CAN). Even

an adjustable heated seat may require dozens of control wires, but with a network only one power and one control cable is needed. The engine management system, lighting assemblies, radio, CD player, and wind screen wipers each have a small controller that talks through the network to the driver's console (although critical engine systems will usually have a separate circuit).

Many household appliances are now being made Internet-ready. In some cases this may mean an actual interface; for example, an Internet fridge that can scan the barcodes of items as you put them in and out and then warn you when items are getting out of date, generate a shopping list of items for you, and even order from your favorite store (Electrolux, 1999). Others have instead, or in addition, connectivity for maintenance purposes, sending usage and diagnostic data back to the manufacturer so that technicians can organize service or repair visits before the appliance fails in some way.

In some ways Internet shopping can also be seen in this light. Whilst at one level it is merely a transfer of data, the ultimate end is that you receive physically the ordered goods. This interaction with the remote physical goods is often two-way as you track its progress, and sometimes its physical location, through a web interface.

Data

Anyone using the web is accessing remote data. Sometimes, data is stored remotely purely for convenience, but often data is necessarily stored remotely

- because it is shared by many remote people;
- because central storage helps maintain control, security, or privacy;
- because it is used by a single user at different locations (e.g., web e-mail);
- because it is too extensive to be stored locally (e.g., large databases and thin client).

In the case of the Web the data is remote because it is accessed by different people at different locations, the author(s) of the material, and all those who want to read it.

Even though the web is quite complex, we may perceive a web page as a single entity, but in fact it exists in many forms (Fig. 14.1). The author of the page will typically have created it offline on his or her own PC. He or she then uploads the page (which effectively means copying it) onto the web server. Any changes the author makes after uploading the page will not be visible to the world until it is next uploaded. When a user wants to see the page and enters a URL or clicks on a link, his browser asks the web server for the file, which is then copied into the browser's memory and displayed to the user. You can tell the browser has a copy as you can disconnect from the Internet and still scroll within the file. If you access the same page again quite soon, your browser may choose to use the copy it holds rather than going back to the web server, again potentially meaning you see a slightly out-of-date copy of the page. Various other things may keep their own cached copies, including web proxies and firewalls.

This story of copied data in various places is not just about the web, but is true to some extent or other of all shared networked data. With people or physical things, we do not expect to have the actual person or thing locally, just a representation. This is equally true for shared data except that the representation is so much like the "real thing" it is far less obvious to the user.

For shared networked data, even the "real thing" may be problematic—there may be no single "golden copy," but instead many variants, all with equal right to be called "the real data."

You don't even escape networking issues if you only access data locally on your own PC; networking issues may still arise if your data is backed-up over the network.

Computation

Sometimes remote computational resources are accessed over the network. The most obvious example of this is large supercomputers. These have enormous computational power, and scientists wishing to use them will often pre-book time slots

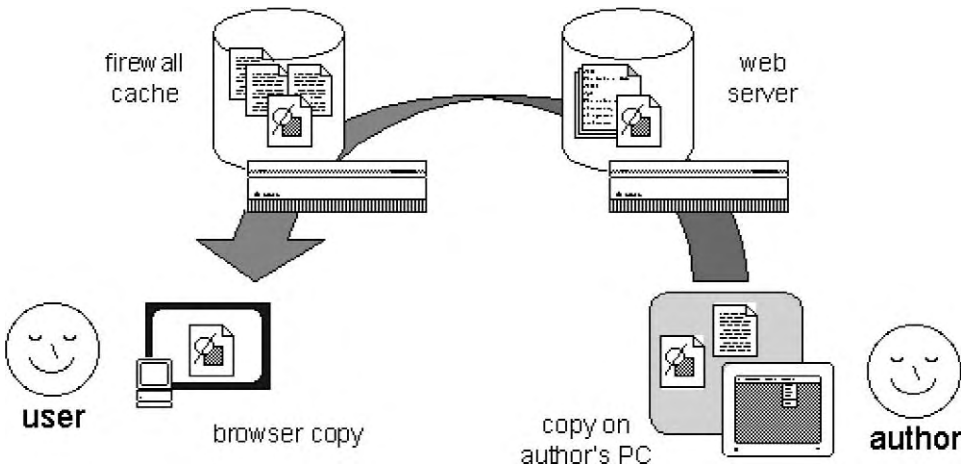


FIGURE 14.1. Copies of a web page in many places.

to perform particularly intensive calculations such as global weather simulations, analysis of chemical structure, stress calculations, and so on. Because these machines are so expensive, programs for them are typically developed on other, less powerful computers and then uploaded over the network when the supercomputer is available.

If the data required as input or output for the calculation is not too great, "fairly simple" means can be used to upload the programs and data. However, some calculations work on large volumes of data; for example, data from microwave readings of the upper atmosphere to probe the ozone hole generate terabytes (millions of millions of bytes) of data per second. High-capacity networks are being created in many countries to enable both high-volume data for this sort of application as well as the expected data required for rich media (Foster, 2000; Foster & Kesselman, 1999; GRID, 2001).

The ease with which data and results can be shipped back and forth across the Internet has enabled the growth of web services, web applications designed to be accessed by other programs supplying services or data. As well as more scientific or heavy commercial uses of these, they have become a standard part of many consumer-oriented applications—for example, "del.icio.us" has a standard API accessible through the web allowing third-party applications to interact with it.

Sometimes calculations need to be performed centrally, not because the central computer is powerful, but because the local device is a computational lightweight. For example, one may want to create a remote analysis package where engineers in the field enter data into a PDA or phone interface, but where complex stress calculations are carried out on a small server back in the office. The data on materials and calculations involved may not be extensive by supercomputer standards, but may still be too much for a handheld device.

Because transporting large volumes of data is not always practical, calculations are often performed where the data is. (In performing any computation program, data and computational engines must all be in the same place. If they aren't together, then one or the other must be moved or copied to bring them together (Ramduny & Dix, 1997)). For example, when you perform a database access, the request for the data is usually transmitted to the database server as an SQL query, for example: "SELECT name, salary FROM payroll WHERE salary > 70000." In principle the complete contents of the payroll database could be downloaded to your PC and the selection of appropriate records carried out locally, however, it would be more costly to transmit the data hence the calculation is effectively transmitted to the database server. In a similar vein, Alexa allows third parties to run programs on their servers through their Web Search Platform (AlexaWSP, 2006); this allows the programs to access a 100-terabyte web crawl that would be impractical (and commercially unacceptable) to transfer to clients.

Even when the volume of data is not large or the frequency of access would make it cost effective to transmit it, security or privacy reasons may prevent the download of data. For example, some datasets are available to search to a limited degree on the web, but charge for a download or CD of the complete dataset. My own hcibook.com site (Dix, Finlay, Abowd, & Beale, 1998) is rather like this, allowing searching of the book's contents online and displaying portions of the text, but not allowing a full download as readers are expected to buy the book!

> For those who haven't come across it, the SETI (the Search for Extra-Terrestrial Intelligence) project is analyzing radio signals from outer space looking for patterns or regularities that may indicate transmissions from an alien civilization. You can download a SETI screensaver that performs calculations for SETI when you are not using your machine. Each SETI screensaver periodically gets bits of data to analyze from the central SETI servers and then returns results. This means that the SETI project ends up with the combined computational resources of many hundreds of thousands of PCs.

Security considerations may also prohibit the distribution of programs themselves if they contain proprietary algorithms. Also, if the source of the program is not fully trusted, one may not want to run these programs locally. The latter is the reason that Java applets are run in a software "sandbox" confining the ability of the applet to access local files and other potentially vulnerable resources.

The Search for Extra-Terrestrial Intelligence (SETI) is an interesting example of remote computation (SETI@home, 2001). Normally, remote computation involves a device of low-computational power asking a central computer to do work for it. In the case of SETI, large calculations are split up and distributed over large numbers of not particularly powerful computers.

The same technique is used in "PC farms." These occur when large numbers of PCs are networked together to act as a form of super-computer. For example, in CERN (the home of the web), data from high-energy collisions may consist of many megabytes of data for each event, with perhaps hundreds of significant events per second (CERN, 2001). The data from each event is passed to a different PC which then performs calculations on the data. When the PC finishes it stores its results and then adds itself back to a pool of available machines.

During coming years we are likely to see both forms of remote computation. As devices become smaller and more numerous, many will become simply sensors or actuators communicating with central computational and data servers (although "central" here may mean one per room, or even one per body). On the other hand, several companies, inspired by SETI, are pursuing commercial ways of harnessing the spare, and usually wasted, computational power of the millions of home and office PCs across the world.

Applications

The existence of networks, particularly the global networks offered by the Internet and mobile phone networks, have made many new applications possible and changed others.

Several of the more major application areas made possible by networks are covered in their own chapters: groupware (chapter 29), online communities (chapter 30), mobile systems (chapter 32), e-commerce (chapter 39), telecommunications (chapter 40) and, of course, the web (chapter 37).

In addition, networking impinges on many other areas. Handheld devices (chapter 32) can operate alone, but are increasingly

able to interact with one another and with fixed networks via wireless networking. Similarly, wearable computers (chapter 33) are expected to be interacting with one another via short-range networks, possibly carried through our own bodies (which makes mobile phones seem positively safe!) and information appliances (chapter 38) will be Internet connected to allow remote control and maintenance. In the area of government and citizenship (chapter 41), terms such as *e-democracy* and *e-government* are used to denote not just the technological ability to vote or access traditional government publications online, but a broader agenda whereby citizens feel a more intimate connection to the democratic process. Of course, education, entertainment, and game playing are also making use of networks.

Throughout the chapter we will also encounter broader issues of human abilities, especially concerned with time and delays, involving aspects of virtually all of part II (human perception, cognition, motor skills, etc.). Also, we will find that networking raises issues of trust and ethics (chapters 65 and 62) and of course the global network increases the importance of culturally and linguistically accessible information and interfaces (chapter 23).

Networking has already transformed many people's working lives, allowing telecommunication, improving access to corporate information whilst on the move, and enabling the formation of virtual organizations. Networks are also allowing whole new business areas to develop, not just the obvious applications in e-shopping and those concerned with web design.

The Internet has forced many organizations to create parallel structures to handle the more direct connections between primary supplier and consumer (disintermediation). This paradoxically is allowing more personalized (if not personal) services and often a focus on customer–supplier and customer–customer communication (Siegal, 1999; Light & Wakeman, 2001). This restructuring may also allow the more flexible businesses to revolutionize their high street (or mall) presence, allowing you to buy shoes in different sizes, or use next-day fitting services for clothes (Dix, 2001b).

The complexity of installing software and the need to have data available anywhere at any time has driven the nascent application service provider (ASP) sector. You don't install software yourself, but instead use software hosted remotely by providers who charge on a usage rather than once-off basis. By storing the data with third parties, an organization can off-load the majority of its backup and disaster-management requirements.

For the individual user, the ubiquity of Internet access for many has enabled many PIM (personal information management) applications such as e-mail, calendars, bookmark lists, and address books. These things that would once have been seen as personal are being not only accessed via the web, but in many cases also shared. These web communities are no longer the province of geeks, but have become part of the day-to-day lives of many, engendering whole new ways of finding out and getting to know, including social bookmarks, blogs, and photologs.

The "personal" device has not become redundant though in this web orientation of applications. As well as being an access point to global services, it is also a potential interaction device for things close by. For example, in an installation by .:thePooch:. (thePooch, 2006) in an arts event, the attendees were encouraged to send SMS texts to Andrine, a huge face projected high on the wall. The texts were analyzed using natural language-processing techniques, and, depending on the content, the face took on different emotions: happy, sad, or shocked (Lock, Allanson, & Phillips, 2003). The cameras in phones are also being used to enable them to be used as location-finding devices (Sarvas, Herrarte, Wilhelm, & Davis, 2004), to enable the embedding of SpotCodes or other visual codes in paper posters (Toye et al., 2004; Semacode, 2006), and for real time manipulation of large public displays (Rohs, Sheridan, & Ballagas, 2004). Some of these applications use local networking such as Bluetooth; others paradoxically use the "global" connectivity through SMS or WAP to enable local interactions.

NETWORKS AS MEDIATORS

Issues and Problems Because of Networks

This section takes as a starting point that an application is networked and looks at the implications this has for the user interface. This is most apparent in terms of timing problems of various kinds. This section is really about when the network is largely not apparent, except for the unintended effects it has on the user.

We'll begin with a technical introduction to basic properties of networks and then see how these affect the user interface and media delivery.

Network Properties

Bandwidth and Compression

The most commonly cited network property is *bandwidth,* how much data can be sent per second. Those who have used dial-up connections will be familiar with 56 K modems, and those with longer memories or who use mobile phone modems may recall 9.6 K modems or less. The "K" in all of these refers to thousands of bits (0/1 value) per second (strictly Kbps), rather than bytes (single characters) that are more commonly seen in disk and other memory sizes. A byte takes eight bits; taking into account a small amount for overhead, you can divide the bits per second by 10 to get bytes per second.

Faster networks between machines in offices are more typically measured in megabits per second (again, strictly Mbps but often just written "M")—for example, the small "telephone cable" Ethernet is rated at either 10 Mbps or 100 Mbps.

As numbers these don't mean much, but if we think about them in relation to real data the implications for users become apparent.

A small word processor document may be 30 Kb (kilobytes). Down a 9.6 K GSM modem this will take approximately half a minute; on a 56 K modem this is reduced to five seconds; for a 10 Mb Ethernet this is 30 milliseconds. A full-screen, web-quality graphic may be 300 Kb, taking five minutes of 9.6 K modem, less than a minute on a 56 K modem, or one-third

Note that I am using the formula:

$$\text{download time } T = \frac{F \times 10}{M}$$

where:
F = size of file in bytes
10 is the number of raw bits per byte
M = modem speed in bits per second

of a second on 10 Mb Ethernet. (Note that these are theoretical minimum times if there is nothing else using the network.)

Rich media, such as sound or video, puts a greater load on again. Raw, uncompressed HI-FI quality sound needs over 200 kilobits per second, and video tens of megabits per second. Happily there are ways to reduce this; otherwise digital AV would be impossible over normal networks.

Real media data has a lot of redundant information: areas of similar color in a picture, successive similar frames in a video, or sustained notes in music. **Compression** techniques use this similarity to reduce the actual amount of data that needs to be sent (e.g., rather than sending a whole new frame of video, it just sends the differences from the last frame). Also, some forms of compression make use of human perceptual limits; for example, MP3 stores certain pitch ranges with greater fidelity than others, as the human ear's sensitivity is different at different pitches (MPEG, 2001); also, JPEG images give less emphasis to accurate color hue than the darkness/lightness (JPEG, 2001). Between them, these techniques can reduce the amount of information that needs to be transferred significantly, especially for richer media such as video. Thus the actual bandwidth and the effective bandwidth, in terms of the sorts of data that are transmitted, may be very different.

Latency and Start-Up

Bandwidth measures how much data can be transferred; *latency* is how long each bit takes. In terms of a highway, bandwidth would be the number of lanes and latency is the time it takes to travel the length of the highway. The latency is due to two factors. The first is the speed of transmission of electricity through wires or light through optical networks. This may seem insignificant, but for a beam of light to travel across the Atlantic would take 20 ms and in practice this hop takes more like 70 ms. For satellite-based communications, the return trip to and from a geostationary satellite takes nearly a second; think about the typical delay you can hear on a transcontinental telephone call. The second factor contributing to latency is that every electronic switch or computer router has to temporarily store and then decide what to do with the signal before passing it on to the next along the chain. Typically this is a more major factor and in practice transatlantic Internet traffic will take nearer 250 ms from source to final destination, most of which is spent in various computer centers at one end or the other.

Latency is made worse by *setup time*. Every time you establish an Internet connection a conversation is established between your computer and the machine hosting the web-server:

"Hello, are you there?"
"Yes, I'm here. What do you want?"
"I'd like to send you some data."
"Great. I'm waiting."
"Okay, here it is, then."

(This is called handshaking.) Each turn in this conversation involves a round trip, network latency on both outward and return paths, and processing by both computers. And this is before the web server proper even gets to look at your request. Similar patterns happen as you dial a telephone call.

Latency and setup time are critical as they often dominate the delay for the user except for very large files or streaming audio/visual media. Early web design advice (given by those concerned about people with a slow connection, but who clearly had never used one!) used to suggest having only as much text as would fit on a single screen. This was intended to minimize the download time. However, this ignores setup times. A long text page doesn't take long to load even on a slow connection, once the connection to the web server has been established. Then it is far faster to scroll in the browser than to click and wait for another small page to load. A similar problem is the practice of breaking large images up into a jigsaw of small pieces. There are valid reasons for this—allowing rollover interaction, or where parts of the image are of different kinds (picture/text)—however, it is also used without such reasons and each small image requires a separate interaction with the server, encountering latency and setup delays.

Jitter and Buffering

Suppose you send letters to a friend every three days and the postal service typically takes two days to deliver letters (the average latency in network terms). Your friend will receive letters every three days, just delayed from when you sent them. Now

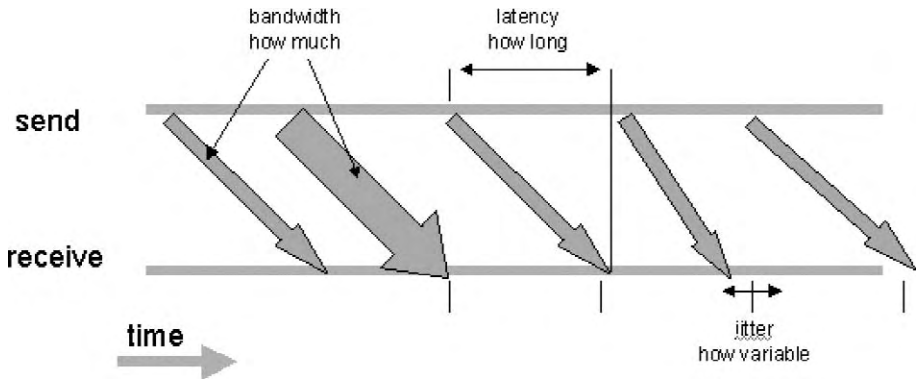


FIGURE 14.2. Bandwidth, latency, and litter.

imagine that the postal system is a little variable: sometimes letters take two days, but occasionally they are faster and arrive the next day and sometimes they are slower and take three days. You continue to send letters every three days, but if a slow letter is followed by a fast one, your friend will receive them only one day apart, if on the other hand a fast letter is followed by a slow one the gap becomes five days. This variability in the delay is called "jitter." (Note that the fast letters are just as problematic as the slow ones; a fast letter followed by a normal-speed one still gives a four-day gap.)

Jitter doesn't really matter when sending large amounts of data, or when sending one-off messages. However it is critical for continuous media. If you just played video frames or sound when it arrived, jitter would mean that the recording would keep accelerating and slowing down (see Fig. 14.3(a) and (b)).

Jitter can be partially alleviated by *buffering*. Imagine that your friend's postman holds back one letter for three days and then starts giving letters to your friend one every third day. If your mail always arrives in exactly two days, the postman will always hold exactly one letter as mail will arrive as fast as he passes it on. If however a letter arrives quickly he will simply hold two letters for a few days and if it is slow he will have a spare letter to give. Your friend's mail is now arriving at a regular rate, but the delay has increased to (a predictable) five days.

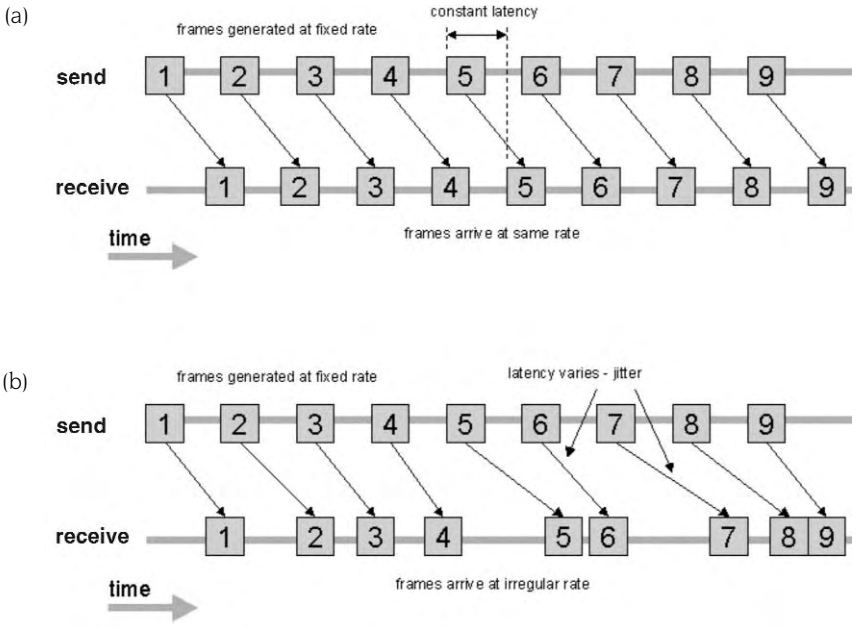


FIGURE 14.3. (a) No jitter, no problem. (b) Jitter causes irregular perception.

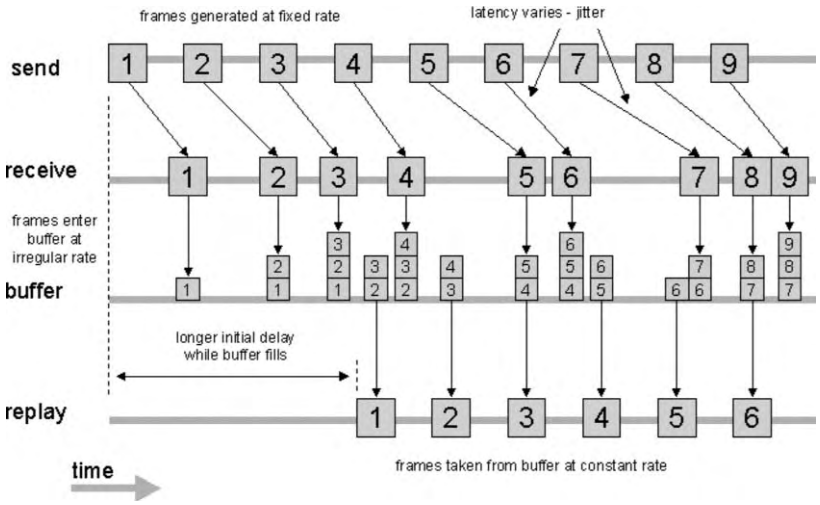


FIGURE 14.4. Buffering smooths jitter, but adds delay.

Buffering in network multimedia behaves exactly the same, holding back a few seconds of audio/video data and then releasing it at a constant rate.

Reliability and Loss, Datagram, and Connection-Based Services

Virtually all networks are designed on the principle that there will be some loss or damage to data en-route. This arises for various reasons; sometimes there is electrical interference in a wire, or the internal computers and routers in the network may have too much traffic to cope with. This is normal and network software is built to detect damaged data and cope with lost data.

Because of this, the lowest layers of a network are assumed to be "lossy." Any data damaged in transit are discarded and computers and hardware en-route can choose to discard data if they get busy. So when one computer sends a packet of data to another it can assume that if the packet of data arrives it will be intact, but it may never arrive at all.

Some network data, in particular certain forms of real-time multimedia data, are deliberately sent in this unreliable, message-at-a-time, form (called "datagrams").

However, it is usually easier to deal with reliable channels, and so higher levels of the network create what are called *connection-based* services on top of the unreliable lower-level service. Internet users may have come across the term TCP/IP. IP is the name of an unreliable low-level service that routes packets of data between computers. TCP is a higher-level, connection-based service built on top of IP. The way TCP works is that the computer wanting to make a connection contacts the other and they exchange a few (unreliable IP) messages to establish the link. Once the link is established, the sending computer tags messages it sends with sequence data. Whenever the receiving computer has all the data up to a certain point it sends an acknowledgement. If the sending computer doesn't get an acknowledgement after a certain time it resends the data (Stevens, 1998, 1999).

With TCP, the receiving computer cannot send a message back when it notices a gap; it has to wait for the sending computer to resend after the timeout. While it is awaiting the resend it cannot process any of the later data. Notice that this means reliability is bought at the price of potential delays.

Quality of Service (QoS) and Reservation

The above properties are not just determined by a raw network's characteristics, such as the length of wires, types of routers, modems, and so on. They are also affected by other traffic and its volume and nature. If 10 PCs are connected to a single 10 Mbps network connection and require high-volume data transfers (perhaps streaming video), then there is only, on average, 1Mbps available for each. If you are accessing a network service that requires transatlantic connections during peak hours, then intermediate routers and hubs in the network are likely to become occasionally overloaded, leading to intermittent packet loss, longer average delays, and more variability in delay, hence jitter. In the past when the capacity of Internet backbones was lower, it was obvious in the United Kingdom when the United States "woke up" as the web ground to a crawl!

For certain types of activity, in particular real-time or streaming rich-media, one would like to be able to predict or guarantee a minimum bandwidth, maximum delay, jitter, and so forth. These are collectively called "quality of service (QoS)" issues (Campbell & Nahrstedt, 1997). Some network protocols allow applications to reserve a virtual channel with guaranteed properties; however, the most common large-scale network, the Internet, does not have such guarantees—it operates solely on a best-endeavor basis. Upgrades to the underlying protocol (called by the catchy name IPv6) allow some differentiation of different types of traffic. This may allow routers to make decisions to favor time-critical data, but it will still not be able to reserve guaranteed capacity. However, in practice the increased capacity of Internet backbones is allowing large-scale Voice-over-Internet services such as Skype with acceptable end-to-end service (FCC, 2006; Skype, 2006).

Encryption, Authentication, and Digital Signatures

Some networks, such as closed office networks, offer no greater worries about security of information than talking together (both are capable of being bugged, but with similar levels of difficulty). However, more open networks, such as the Internet or phone networks, mean that data is traveling through a third party and public infrastructure to get to its recipients. Increasing use of wireless devices also means that the data sent between devices is more easily able to be monitored or interfered with by third parties. One option is to only use physically secure networks, but for economic reasons this is often not an option. Furthermore, solutions which do not rely on the network itself being secure are more robust. If you rely on, for example, a private dedicated line between two offices and assume it is secure, then if someone does manage to tap into it, all your interoffice communication is at risk.

The more common approach now is to assume the networks are insecure and live with it. This gives rise to two problems:

Secrecy—how to stop others from seeing your data
Security—how to make sure data is not tampered with

The first problem is managed largely by encryption methods, ensuring that even if someone reads all your communications they cannot understand them (Schneier, 1996). The "https" in some URLs is an example of this, denoting that the communication to the web server is encrypted.

The second problem, security, has various manifestations. Given communications are via a network, how do you know that you are talking to the right person/machine? Authentication mechanisms deal with this. In various ways they allow one machine to verify (usually by secret information that can only be known by the true intended party) that it is talking to the right party.

Even if you know that you are talking to the right person/machine, how do you know that the data you receive hasn't been changed? This is like receiving a signed letter, but, unbeknownst to you, someone has added some lines of text above the signature—although it really comes from the person you think, the message is not as was sent. If data is being encrypted

then this may often implicitly solve this problem as any tampered data is uninterpretable by a third party, who therefore cannot alter it in a meaningful way.

If secrecy is not an issue, however, encryption is an unnecessary overhead and instead digital signatures generate a small data block that depends on the whole of the message and secret information known to the sender. It is possible for the recipient to verify that the signature block corresponds to the data sent and the person who is supposed to have sent it. One example of this are "signed applets" in which the Java code is digitally signed so that you can choose to only run Java programs from trusted parties.

UI Properties

Network Transparency

One of the goals of many low-level network systems is to achieve transparency; that is, to make it invisible to the user where on the network a particular resource lies. When you access the web you use the same kind of URL and same kind of interface whether the web server is in Arizona, Australia, or Armenia. I know that when I'm at home and send an e-mail between two machines less than two meters apart, the message actually goes all the way across the Atlantic and back—but this is only because I have quite a detailed understanding of the computers involved. As a user I press "send mail" on one machine and it arrives near-instantaneously on the other.

Although network transparency has many advantages to the user—you don't care about routes through the network and so on—there are limits to its effectiveness and desirability. Some years ago I was at a Xerox lab in Welwyn Garden City in the United Kingdom. Randy Trigg was demonstrating some new features of Notecards (an early hypertext system; Halasz, Moran, & Trigg, 1987). The version was still under development and every so often would hit a problem and a LISP debugger window would appear. After using it for a while, it suddenly froze—no debugger window, no error message; it just froze. After a few seconds of embarrassment, Randy hit a control key and launched the debugger. A few minutes of frantic scanning through stack dumps, program traces, and so on, and the reason became clear to him. He had demonstrated a feature that he had last used on his workstation at Palo Alto. The feature itself was not at fault, but required an obscure font that he had on his own workstation, but not on the machine there in Welwyn. When Notecards had requested the font the system might have thrown up an error window, or substituted a similar font. However, in the spirit of true network transparency, the location of the font should not matter; having failed to find it on the local machine, it proceeded to interrogate machines on the local network to see if they had it, and then proceeded to scan the Xerox UK network and world network. Eventually, if we had waited long enough, it would have been found on Randy's machine in Palo Alto. Network transparency rarely extends to timing!

Transparency has also been critiqued for CSCW purposes (Mariani & Rodden, 1991). It may well be very important to users where resources and people are. For mobile computing also, an executive takes a laptop on the plane only to discover that the files needed are residing on a network fileserver rather than on the machine itself. If the interface hides location, how can one predict when and where resources will be available?

Later in this chapter, we will discuss recent work where the presence of intermittent connections, limited range, and variable signal strength is being used as a deliberate feature in interfaces.

Delays and Time

As is evident, one of the issues that arises again and again when considering networks is time. How long are the delays, how long to transfer data, and so on? Networking is not the only reason for delays in applications, but is probably one of the most noticeable—the web has often been renamed the "World-Wide Wait." There is a long-standing literature on time and delays in user interfaces. This is not as extensive as one might think, however, largely because for a long time the prevailing perception in the HCI community was that temporal problems would go away (with some exceptions), leading to what I called the "myth of the infinitely fast machine" (Dix, 1987).

One of the earlier influential papers was Ben Shneiderman's review of research findings on delays (Shneiderman, 1984), mainly based on command-line interfaces. More recently there have been a series of workshops and special journal issues on issues of time, sparked largely by web delays (Johnson & Gray, 1996; Clarke, Dix, Ramduny, & Trepess, 1997; Howard & Fabre, 1998).

Three main timescales are problematic for networked user interfaces:

100 ms	Feedback for hand-eye coordination tasks needs to be less than 100–200 milliseconds to feel fluid. This is probably related to the fact that there are delays of this length in our motor-sensory system anyway. For aural feedback, the timescales are slightly tighter again.
1 second	Timescale for apparent cause–effect links, such as popping a window after pressing a button. If the response is faster than this the effect seems "immediate." This is related to a period of about 1 second that the brain regards as "now."
5–10 seconds	Waits longer than this engender annoyance and make it hard to maintain task focus. This may be related to short-term memory decay.

The 100 ms time is hard to achieve if the interaction involves even local network traffic. The 1 second time is usually achievable for local networks (as are assumed by X-Windows systems), but more problematic for long-haul networks. The 5–10 second time is in principle achievable for even the longest transcontinental connections, but, when combined with bandwidth limitations or overload of remote resources, may become problematic. This is especially evident on web-based services where the delay between hitting a link and retrieving a page (especially a generated page) may well exceed these limits, even for the page to begin to draw.

The lesson for UI designers is to understand the sort of interaction required and to ensure that parts of the user interface are located appropriately. For example, if close hand-eye coordination is required, it must run locally on the user's own machine—in the case of the web in an applet, in JavaScript code, or so on. If the nature of the application is such that parts of the application cannot reside close enough to the user for the

type of interaction required then, of course, one should not simply have a slow version of (for example) dragging an icon around, but instead change the overall interaction style to reflect the available resources.

Two of the factors that alleviate the effects of longer delays are predictability of the delay and progress indicators. Both give the user some sense of control or understanding over the process, especially if users have some indication of expected delays before initiating an action (Johnson, 1997). The many variable factors in networked systems make predicting delays very difficult, increasing the importance of giving users some sense of progress. The psychological effect of progress indicators is exploited (cynically) by those web browsers that have progress bars that effectively lie to the user, moving irrespective of any real activity. (Try unplugging a computer from the network and attempting to access a web page; some browsers will hit 70% on the progress bar before reporting a problem.) Other network applications use recent network activity to predict remaining time for long operations (such as large file downloads). Other solutions include generating some sort of intermediate low quality or partial information while the full information is being generated or downloaded (e.g., progressive image formats or splash pages of Flash movies).

For virtual reality using head-mounted displays, as well as hand-eye coordination tasks, we also have issues of the coordination between head movements and corresponding generated images. The timescales here are even tighter as the sensory paths are faster within our bodies, hence less tolerant of external delays. The brain receives various indications of movement: the position and changes of neck and related muscles, the balance sensors in the inner ear, and visual feedback. Delays between the movement of the generated environment and head movement lead to dissonance between these different senses and have an effect rather like being at sea, with corresponding disorientation and nausea. Also, any delays reduce the sense of immersion—being there within the virtual environment. Early studies of virtual reality (VR) showed that users' sense of immersion was far better when they were given very responsive wire frame images than if they were given fully rendered images at a delayed and lower frame rate (Pausch, 1991).

Coping Strategies

People are very adaptable. When faced with unacceptable delays (or other user interface problems) users develop ways to work around or ameliorate the problem—coping strategies. For example, web users may open multiple windows so that they can view one page whilst reading another (McManus, 1997) and users of "telnet" for remote command line interfaces may type "test" characters to see whether the system has any outstanding input (Dix, 1994).

Coping strategies may hide real problems, so it is important not to assume that just because users do not seem to be complaining or failing that everything is all right. On the other hand, we can also use the fact that users are bright and resourceful by building interface features that allow users to adopt coping strategies where it would be impossible or impractical to produce the interface response we would like. For example, where we expect delays we can ensure that continual interaction is not required (by perhaps amassing issues requiring user attention in a "batch" fashion), thus allowing users to more easily multitask. Unfortunately, this latter behavior is not frequently seen—the "myth" lives on and most networked programs still stop activity and await user interaction whenever problems are encountered.

Timeliness of Feedback/Feedthrough, Pace

Although feedback is one of the most heavily used terms in HCI, we may often ignore the complex levels of feedback when dealing with near-instantaneous responses of GUI interfaces. In networked systems with potentially long delays we need to unpack the concept. We've already discussed some of the critical timescales for feedback. For hand-eye coordination, getting feedback below the 100 ms threshold is far more important than fidelity; quickly moving wire frames or simple representations are better than dragging an exact image with drop shadow.

For longer feedback cycles, such as pressing a button, we need to distinguish

- **Syntactic feedback**—that the system has recognized your action
- **Intermediate feedback**—that the system is dealing with the request implied by your action (and, if possible, progress towards that request)
- **Semantic feedback**—that the system has responded and the results were obtained

The direct-manipulation metaphor has led to identification and hence confusion between these levels, and many systems provide little in the way of syntactic or intermediate feedback, relying solely on semantic feedback. In networked systems where the semantic feedback includes some sort of remote resource, it is crucial to introduce specific mechanisms to supply syntactic and intermediate feedback; otherwise, the system may simply appear to have ignored the user's action (leading to repeated actions with potentially unforeseen consequences) or even frozen or crashed.

This also reminds us of a crucial design rule for slow systems: Wherever possible, make actions idempotent—that is, that invoking the same action twice should, where possible, have the same effect as a single action. This means that the "try again" response to a slow system does not lead to strange results.

For collaborative systems or those involving external or autonomous resources (e.g., remote controlled objects, environmental sensors, software agents), we must also consider feedthrough. Feedback is experiencing the effect of one's own actions; *feedthrough* is the effect of one's own actions on other people and things and experiencing the effects of their actions oneself. For example, in an online chat system, you type a short message, press "send," and your message appears in your transcript (feedback); then, sometime later, it also appears in the transcript of the other chat participants (feedthrough).

Feedback is needed to enable us to work out whether the actions we have performed are appropriate, hence (typically) needs to be much quicker than feedthrough responses. This is fortunate as feedthrough by its very nature usually requires network transmission and ensuing delays. The exception to the rule that feedthrough can afford to be slower is where the users

are attempting to perform some close collaborative task (e.g., positioning some items using direct manipulation) or where there is a second, fast communication channel (e.g., on the telephone, User A says to User B, "See the red box," but the relevant item hasn't appeared yet on User B's screen).

Potentially more important to users of collaborative systems than bandwidth or even raw delays is *pace,* the rate at which it is possible to interact with a remote resource or person. This is partly determined by lower level timings, but is also heavily influenced by interface design. For example, you know that someone is sitting at their desk and send them an urgent e-mail. The time that it takes to get a response will be hardly affected by the raw speeds between your machine and your colleague, and more determined by factors such as how often the e-mail client checks the server for new e-mail and whether it sounds an alert when new e-mail arrives, or simply waits there until your colleague chooses to check the in-box.

Race Conditions and Inconsistent Interface States

Alison and Brian are using an online chat program.

Alison writes, "It's a beautiful day. Let's go out after work," and then begins to think about it.
Brian writes, "I agree totally," and then has to leave to go to a meeting.
At almost the same time Alison writes, "Perhaps not; I look awful after the late party."

Unfortunately the messages are so close to simultaneous that both Alison and Brian's machines put their own contribution first, so Alison sees the chat window as in Fig. 14.5.a and Brian sees it as in Fig. 14.5.b. Brian thinks for a few moments, and then writes, "No, you look lovely as ever," but unfortunately Alison never sees this as she took one look at Brian's previous remark and shut down the chat program.

This type of incident where two events happen so close together that their effects overlap is called "a race condition." Race conditions may lead to *inconsistent states* for users as in this example, or may even lead to the software crashing. Although in principle race conditions are possible however fast the underlying network, the likelihood of races occurring gets greater as the network (and other) delays get longer.

Even some of the earliest studies in collaborative systems have shown the disorienting effects of users seeing different views of their shared information space, even when this is simply a matter of seeing different parts of the same space (Stefik, Bobrow, Foster, Lanning, & Tatar, 1987).

Consistency becomes an even greater problem in mobile systems where wireless connections may be temporarily lost, or devices may be unplugged from fixed networks whilst on the move. During these periods of *disconnection,* it is easy for several people to be updating the same information, leading to potential problems when their devices next become network-connected.

Later in this chapter, we will discuss mechanisms and algorithms that can be used to maintain consistency even when delays are long and race conditions are likely to occur.

Awareness

Returning to Alison and Brian, after Brian has typed his response he may not know that Alison hasn't seen his second contribution.

Awareness of who is around and what they are doing is a major issue in CSCW (see Dourish & Bellotti (1992) and McDaniel & Brinck (1997)). It has various forms:

- Being able to tell easily, when you want to know, what other people are doing;
- Being made aware (via alerts, very salient visual cues, etc.) when significant events occur (e.g., new user arrives, someone makes a contribution, etc.);
- Having a peripheral awareness of who is around and what they are up to.

Awareness isn't just about other people. In any circumstance where the environment may change, but not through your own direct action, you may need to know what the current state is and what is happening. This is not confined to networked applications, but applies to any hidden or invisible phenomena, for example, background indexing of your hard-disk contents. In networked applications, anything distant is invisible unless it is made visible (audible) in the interface.

One of the earlier influential experiments to demonstrate the importance of peripheral awareness was ArKola (Gaver, Smith, & O'Shea, 1991). This was a simulated bottling factory

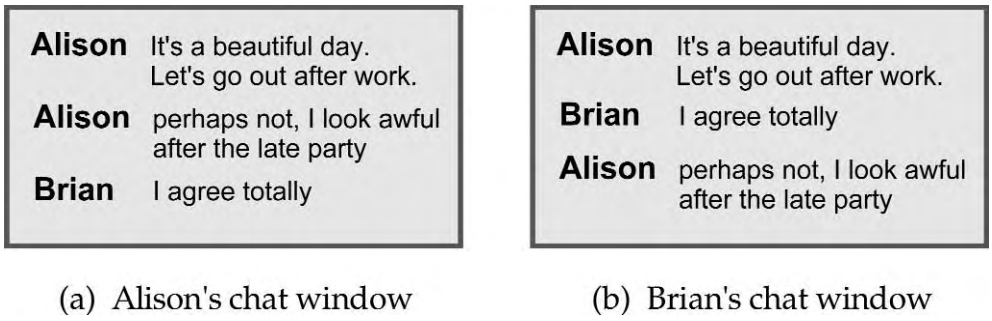


FIGURE 14.5. Consistency breakdown.

where two people worked together to maintain the factory, supplying, maintaining, and so on. The participants couldn't see the entire factory at once, so relied on the sounds produced to be aware of its smooth running or if there are any problems. For example, the sound of breaking glass might suggest that the end of the production line has run out of crates, but if it immediately stopped, one would assume that the other participant had sorted the problem out.

The numerous forms of shared video and audio spaces are another example of this. Several people, usually in distant offices, establish long-term, always-on audio, video, or audio-video links between their offices (Buxton & Moran, 1990; Olson & Bly, 1991). Sometimes these are used for direct communication, but most of the time they just give a peripheral awareness that the other person is there and the sort of activity he or she is doing. This can be used for functional purposes (e.g., knowing when the other person is interruptable), but also for social purposes—feeling part of a larger virtual office. Other systems have allowed larger numbers of (usually deliberately low quality and so less intrusive) web cam views of colleagues' offices and shared areas (Roussel, 1999, 2001). The aim is the same: to build social cohesion, to allow at-a-glance reading of one another's situation, and to promote "accidental" encounters.

A form of awareness mechanism is now common on the web with "buddy lists" that tell you when friends are online (ICQ, 2001). I still know of no examples of rich media experiments in a domestic environment, for example a virtual kitchen-share with your elderly mother in Minnesota. However, various forms of domestic sharing via the Internet are becoming more common, including the early Casablanca project at Interval (Hindus, Mainwaring, Leduc, Hagström, & Bayley, 2001) using shared electronic sketchpads in the home, and more recent projects (CASIDE, 2005; Taylor, Izadi, Swan, harper, & Buxton, 2006). Some of these applications are fairly standard "computer" interfaces; some are soft surveillance, such as monitoring an elderly relative; however, there is also a stream of research looking at more intimate ways of sharing presence, including pads that glow or warm when a loved one touches them far away or Jenny Tillotson's "Scent Whisper," a pair of Internet-connected brooches which emit a pleasant scent when the loved one whispers on the other end (Tillotson, 2005).

Trying to capture all this information within a computer display can be distracting and can use up valuable screen space—and of course assumes that the computer is there and on. For this reason, several projects have looked at ambient interfaces, which in various ways make the physical environment reflect the virtual. These interfaces monitor various events in the electronic worlds and then change things in the physical environment: lights on the wall, moving strings hung from the ceiling, or even a shaking pot plant (Lock, Allanson, & Phillips, 2000). Again, this is not fundamentally limited to networked environments, but is of course not very useful when the relevant activity is close at hand anyway.

The other side of this is finding out what people are doing in order to signal this to others. For computer activity—are you logged on, have you been typing recently, and what web page are you viewing? —this is in principle available, although the various layers of software may make it hard for an awareness service to discover. For non-computer aspects this is more problematic—are you in the room, busy, with other people?—and may require a range of sensors in the environment, ultrasound, video, and so on, with corresponding privacy issues (see, for example, Bellotti (1993)). Monitoring of everyday objects is another way to achieve this; for example, one experiment used electronic coffee cups with sensors to tell when they were picked up and moved around (Gellersen, Beigl, & Krull, 1999). As more and more devices become networked, it may be that we don't need special sensors, just use the combined information from those available, although the privacy issues remain.

In collaborative virtual-reality environments, knowing that other people are around (as avatars) is as important as in a physical world, but harder due to limited senses (usually just vision and sound) and limited field of view. Furthermore, there are computational costs in passing information such as audio or even detailed positional information around the network when there are tens, hundreds, or thousands of users. Various spatial models have been developed to analyze and implement the idea of proximity in virtual space (Benford, Bowers, Fahlen, Mariani, & Rodden, 1994; Rodden, 1996; Sandor, Bogdan, & Bowers, 1997; Dix et al., 2000). These seek to formalize concepts of (a) where your focus of attention is within the virtual world and thus whether you require full quality audio and visual representation of others, (b) broader areas where you would expect some peripheral awareness where potentially degraded information can be used, and (c) those parts of the space for which you need no awareness information.

Media Issues

When describing the intrinsic network properties, issues for continuous media were mentioned several times. This is because, with the possible exception of close hand-eye coordination tasks, continuous media put some of the tightest requirements on the underlying networks.

Interactive Conversation and Action

Most demanding of all are audio-visual requirements of interactive conversation. Anyone who has had a transcontinental telephone conversation will have some feeling for the problems a delay of a second or two can cause. While actually speaking, the delays are less significant; however, turn-taking becomes very problematic. This is because the speaker in a conversation periodically (and subconsciously) leaves short (200–300 ms) gaps in the flow of speech. These moments of silence act as entry points for the other participant, who is expected to either acknowledge with a "go on" sound, such as "uhm" or perhaps a small nod of the head, or break in with his own conversation. Entries at other points would be seen as butting in and rude, and lack of feedback responses can leave the speaker uncertain as to the listener's understanding. The 200–300 ms is again almost certainly related to the time it takes for the listener's sensory system to get the relevant aural information to the brain and for it to signal the relevant nod, acknowledgement, or start to speak. Clearly our conversational system is finely tuned to the expected intrinsic delays of face-to-face conversation.

When network delays are added it is no longer possible to respond within the expected 200–300 ms window. The speaker

therefore gets no responses at the appropriate points and it is very hard for the listener to break into the flow of speech without appearing rude (by the time they hear the gap and speak, the speaker has already restarted). Some telephone systems are half-duplex; that is, they only allow conversation in one direction at a time, which means that the various vocalizations ("uhu," "hmm," and so on) that give the speaker feedback will be lost entirely while the speaker is actually talking. It is not uncommon for the speaker to have to resort to saying, "Are you there?" due to a loss of sense of presence.

These effects are similar whether one is dealing with pure audio stream (as with the telephone), video streams (as with desktop conferencing), or distributed virtual environments. One VR project in the United Kingdom conducted all of its meetings using a virtual environment in which the participants were represented by cuboid robot-like avatars (called "blockies") (Greenhalgh, 1997). The project ended with an online virtual party. As the music played the participants (and their avatars) danced. Although clearly enjoying themselves, the video of the party showed an interesting phenomenon. Everyone danced alone. There are various reasons for this; for example, it was hard to determine the gender of a potential dancing partner. However, one relates directly to the network delays. Although everyone hears the same music they all were hearing it at slightly different time, furthermore, the avatars for other people will be slightly delayed from their actual movements. Given popular music rhythms operate at several beats per second, even modest delays means that your partner appears to dance completely out of time! In more recent work, predictive algorithms have been used to create "ghost" figures showing the "best guess" location of people and things in collaborative virtual environments (Gutwin et al., 2004).

Reliability

As well as delays, we noted previously that network connections may not always be reliable—that is, information may be lost. Video and audio streams behave very differently in the presence of dropped information. Imagine you were watching a film on a long air flight. The break in the sound when the pilot makes an announcement is much more difficult than losing sight of the screen for a moment or two as the passenger in front stands up. At a smaller scale a fraction-of-a-second loss of a few frames of video just makes the movie seem a little jerky; a smaller loss of even a few tens of milliseconds of audio signal would make an intrusive click or distortion. In general, reliability is more important for audio than video streams and where resources are limited it is typically most important to reserve the quality of service for the audio stream.

Sound and Vision

Why is it that audio is more sensitive than video? Vision works (largely) by looking at a single snapshot. Try walking around the room with your eyes shut, but opening them for glances once or twice a second. Apart from the moment or two as your eyes refocus you can cope remarkably well. Now turn a radio on with the sound turned very low and every second turn the sound up for a moment and back to silent—potentially an interesting remixing sound, but not at all meaningful. Sound more than vision is about change in time. Even the most basic sounds, pure tones, are measured in frequencies, how long between peaks and troughs of air pressure. For more complex sounds the shape of the sound through time—how its volume and frequency mix changes—are critical. For musical instruments it is hard to hear the difference between instruments if they are playing a continuous note, but instantly differentiable by their **attack**, how the note starts. (To get some idea of the complexity of sound see the review in Mitsopoulos' thesis (Mitsopoulos, 2000), and for an insight into the way different senses affect interaction see my AVI'96 paper (Dix, 1996).)

Compression

As we discussed earlier, it is possible to produce reliable network connections, but this introduces additional delays. Compression can also help by reducing the overall amount of audio-visual data that needs to be transmitted, but again may introduce additional delays. Furthermore, simple compression algorithms require reliable channels (both kinds of delays). Special algorithms can be designed to cope with dropped data, making sure that the most important parts of the signal are replicated or "spread out" so that dropped data leads to a loss in quality rather than to interruption.

Jitter

As noted previously, jitter is particularly problematic for continuous media. Small variations in delay can lead to jerky video playback, but is again even worse for audio streams. First of all, a longer than normal gap between successive bits of audio data would lead to a gap in the sound, just like dropped data. And perhaps even more problematic, what do you do when subsequent data arrives closer together—play it faster? Changing the rate of playing audio data doesn't just make it jerky, but also changes the frequency of sound, rendering it meaningless.

(*Aside:* Actually, you can do some quite clever things by digitally speeding up sound but not changing its frequency. These are not useful for dealing with jitter, but can be used to quickly overview audio recordings, or catch up on missed audio streams (Arons, 1997; Stifelman, Arons, & Schmandt, 2001).)

For real-time audio streams, such as video conferencing or Voice over IP (VoIP), it is hard to do anything about this and the best one can do is drop late data and do some processing of the audio stream to smooth out the clicks this would otherwise generate.

Broadcast and Prerecorded Media

Where media is prerecorded or being broadcast, but where a few seconds delay are acceptable, it is possible to do far better. Recall that in several places we saw that better quality can be obtained if we are prepared to introduce additional delays.

If you have used streaming audio or video broadcasts, you will know that the quality is quite acceptable and doesn't have many of the problems described above. This is partly because of efficient compression, meaning that video is compressed to a

fraction of a percent of its raw bandwidth and so can fit down even a modem line. However, this would not solve the problems of jitter. To deal with this, the player at the receiving end buffers several seconds of audio-visual data before playing it back. The buffering irons out the jitter, giving continuous quality.

Try it out for yourself. Tune onto a radio channel and simultaneously listen to the same broadcast over the Internet with streamed audio. You'll clearly hear up to a minute delay between the two.

Fast and free Internet connections have enabled the sharing and distribution of high-quality media for storing and playing later, initially through illegitimate file-sharing including the pre-lawsuit Napster and many current peer–peer applications, but also increasingly through paid-for services such as the current Napster and Apple iTunes (Wikipedia contributors, 2006). As well as forcing existing media publishers to revisit their business models, the rise of web-distributed MP3 and Podcasting has enabled would-be artists and broadcasters to bypass the traditional distribution channels.

Public Perception: Ownership, Privacy, and Trust

One of the early barriers to consumer e-commerce has been distrust of the transaction mechanisms, especially giving credit card details over the web. Arguments that web transactions are more secure than phone-based credit-card transactions or even using a credit card at your local restaurant (which gets both card number and signature) did little to alleviate this fear. This was never as major a barrier in the United States as it was, for example, in Europe, but across the world has been a concern, slowing down the growth of e-shopping (or really e-buying, but that is another story (Dix, 2001a)).

It certainly is the case that transactions via secure channels can be far more secure than physical transactions, in which various documents can be stolen or copied en-route and are in a format much more easy to exploit for fraud. However, knowing that a transaction is secure is more than the mechanisms involved, it is about human trust: Do I understand the mechanisms well enough, and the people involved well enough, to trust my money to it?

In fact, with a wider perspective, this distrust is very well founded. Encryption and authentication mechanisms can ensure that I am talking to a particular person or company and that no one else can overhear. But how do I know to trust that person? Being distant means I have few of the normal means available to assess the trustworthiness of my virtual contact. In the real world I may use the location and appearance of a shop to decide whether I believe it will give good service. For mail order goods I may use the size, glossiness, and publication containing an advertisement to assess its expense and again use this to give a sense of quality of the organization. It is not that these physical indicators are foolproof, but that we are more familiar with them. In contrast, virtual space offers few intrinsic affordances. It is easy and quite cheap to produce a very professional web presence, but it may be little more than a facade. This is problematic in all kinds of electronic materials, but is perhaps most obvious when money is involved.

Even if I trust the person at the other end, how do I know whether the network channel I am using is of a secure kind? Again, the affordances of the physical world are clear. In a closed office, the open street, or in a bar frequented by staff of a rival firm, we will say different things depending on the perceived privacy of the location. In the electronic world we rely on "https" at the beginning of a URL (how many ordinary consumers know what that means?), or an icon inserted by the e-mail program to say a message has been encrypted or signed. We need to trust not only the mechanisms themselves, but also the indicators that tell us what mechanisms are being used (Millett, Friedman, & Felten, 2001; Fogg et al., 2001).

In non-financial transactions, issues of privacy are also critical. We've already seen several examples where privacy issues occur. As more devices become networked, especially via wireless links, and our environment and even our own bodies become filled with interlinked sensors, issues about who can access information about you become more significant. This poses problems at a technical level—ensuring security between devices, at an interface level—being able to know and control what or who can see specific information; and at a perception level, believing in the privacy and security of the systems. There are also legal implications. For example, in the United Kingdom in 2001 it was illegal for mobile telecoms operators to give location information to third parties (Sangha, 2001).

The issue of perception is not just a minor point, but perhaps the dominant one. Networks, and indeed computer systems in general, are by their nature hidden. We do not see the bits traveling down the wires or through the air from device to device, but have to trust the system at even the most basic level. As HCI specialists we believe ourselves a little above the mundane software engineers who merely construct computer systems, as we take a wider view and understand that the interaction between human and electronic systems has additional emergent properties and that it is this complete socio-technical unit which achieves real goals. For networked systems, this view is still far too parochial.

Imagine if the personal e-mail of millions of people was being sucked into the databanks of a transnational computer company, and only being released when accessed through the multinational's own web interface. The public outcry! Imagine Hotmail, Yahoo mail, and so on. How is it that, although stored on distant computers, perhaps half the world away, millions of people feel that it is "their" mailbox and trust the privacy of web mail more than, perhaps, their organization's own mail system. This feeling of ownership of remote resources is more than the technology that protects the security of such systems, it is a cultural phenomenon and a marketing phenomenon. The web mail "product" is not just technology, or interface, but is formed by every word that is written about the product in advertisements, press releases, and media interviews (Dix, 2001a).

NETWORKS AS SUBJECTS

Understanding and Managing Networks

When using a networked application, you don't really care what kind of network is being used, whether the data is sent over copper wires, fiber optic, microwave, or satellite. All you care

about is that the two ends manage to communicate and the effects any of the above have on the end-to-end network properties such as bandwidth discussed in the previous section. However, there are times when the network's internals can't be ignored.

Those involved in installing or managing networks need to understand the internal workings of the network in order to optimize performance and find faults. For ordinary users, when things go wrong in a networked application, they effectively become a network manager and so understanding something of the network can help them to deal with the problem. Even when things are working, having some awareness of the current state of the network may help one predict potential delays, avoid problems, and minimize costs. In some cases this can even be used as a positive part of the interactive experience.

We'll start this section by looking at some of the technical issues that are important in understanding networks. This parallels earlier information in this chapter, but is focused on the internal properties of the network. We'll then look at the interface issues for those managing networks and the ways in which interfaces can make users aware of critical network states. Finally, we'll look briefly at the way models of networks can be used as a metaphor for some of the motor and cognitive behaviors of humans.

Network Models

Layers

Networking is dominated by the idea of layers; lower levels of the network offer standard interfaces to higher levels so that it its possible to change the details of the lower level without changing the higher levels. For example, imagine you are using your PDA to access the Internet whilst in a train (see Fig. 14.6). The web browser establishes a TCP/IP connection to the web server, requests a web page, and then displays it. However, between your PDA and the web server the message may have traveled through an infrared link to your mobile phone, which then used a cell-based radio to send it to a mobile-phone station, then via a microwave link to a larger base station onto a fiber optic telephone backbone, and via various copper wires to your ISP's modem bank. Your ISP is then connected into another fiber optic Internet backbone and eventually via more fiber optic, copper, and microwave links to the web server. To complicate things even further, it may even be that the telephone and Internet backbones may share the same physical cabling at various points. Imagine if your poor PDA had to know about all of this.

In fact, even your PDA will know about at least five layers:

- Infrared—how the PDA talks to the phone
- Modem—how the PDA uses the phone to talk to your ISP
- IP—how your PDA talks to the web-server computer
- TCP—how data is passed as a reliable connection-based channel between the right program on your PDA and the web-server computer
- HTTP—how the browser talks to the web server

Each of these hides most of the lower levels, so your browser needs to know nothing about IP, the modem, or the infrared connection whilst accessing the web page.

The nature of these layers differs both between different types of network (for example, WAP) for sending data over mobile phones and devices; it has five defined layers (Arehart et al., 2000) while the ISO OSI reference model has seven layers (ISO/IEC7498, 1994).

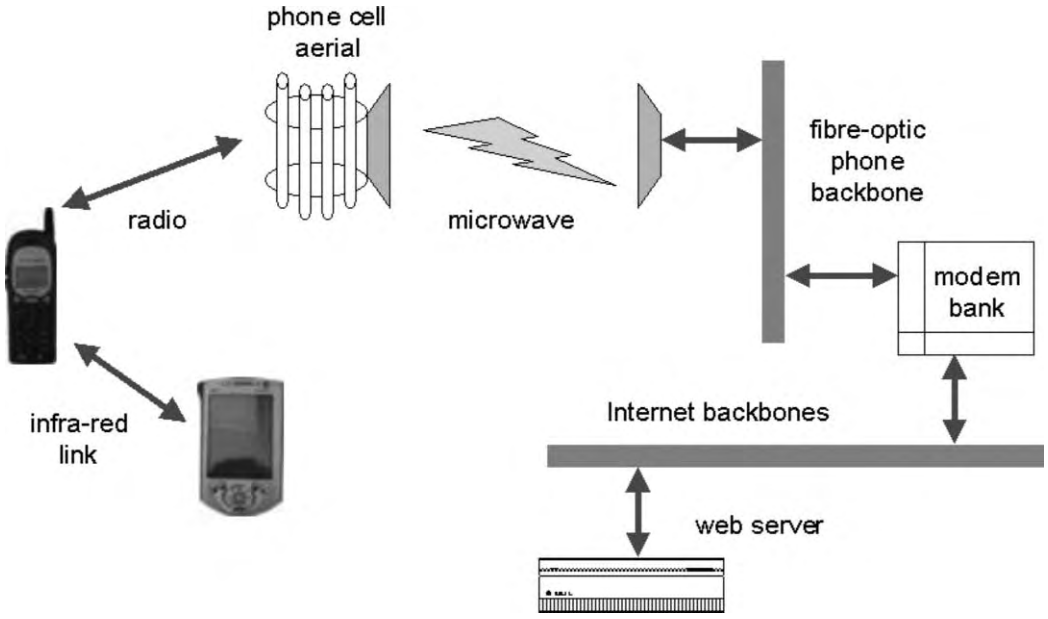


FIGURE 14.6. The long path from PDA to web.

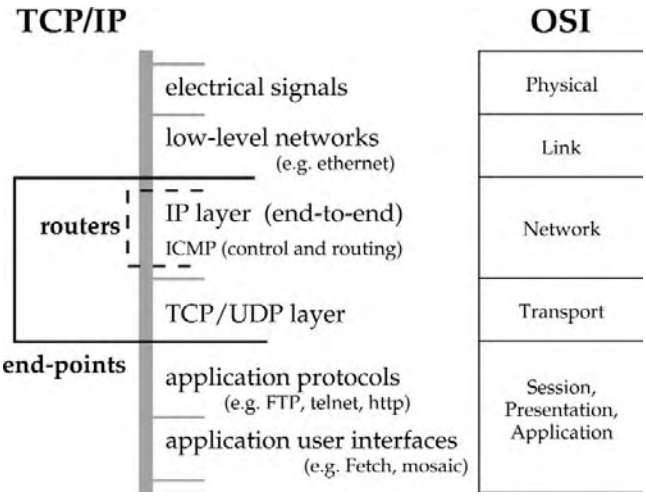


FIGURE 14.7. OSI 7 layers and TCP/IP.

mail program says	SMTP server replies
	220 mail.server.net ESMTP
HELO mypc.mydomain.com	
	250 mail.server.net Hello mypc.mydomain.com[100.0.1.7], pleased to meet you
MAIL From:<myself@mydomain.com>	
	250 <myself@mydomain.com>... Sender ok
RCPT To: <a-friend@theirdomain.com>	
	250 <a-friend@theirdomain.com>... Recipient ok
DATA	
	354 Enter mail, end with "." on a line by itself
first line of message	
..dotty 2nd line	
last line	
.	
	250 KAA24082 Message accepted for delivery
QUIT	
	221 mail.server.net closing connection

FIGURE 14.8. Protocol to send e-mail via SMTP.

Protocols

Systems at the same layer typically require some standard language to communicate. This is called a *protocol*; for higher levels this may be quite readable. For example, to send an Internet e-mail message your mail program connects to an SMTP server (a system that relays messages) using TCP/IP and has the exchange shown in Fig. 14.8.

At lower levels data is usually sent in small **packets**, which contain a small amount of data plus header information saying where the data is coming from, where it is going to and other bookkeeping information such as sequence numbers, data length, and so forth.

Even telephone conversations, except those using pre-digital exchanges, are sent by chopping up your speech into short segments at your local telephone exchange, sending each segment as a packet and then reassembling the packets back into a continuous stream at the other end (Stevens, 1998, 1999).

Internetworking and Tunneling

This layering doesn't just operate within a particular network standard, but between different kinds of networks too. The Internet is an example of an internet (notice lowercase "i") that is a network that links together different kinds of low-level networks. For example, many PCs are connected to the Internet via an Ethernet cable. Ethernet sends its own data in packets like those of Fig. 14.9. The Internet protocol (IP) also has packets of a form like Fig. 14.9. When you make an Internet connection via Ethernet, the IP packets are placed in the data portion of the Ethernet packet, so you get something a bit like Fig.14.10.

This placing of one kind of network packet inside the data portion of another kind of network is also used in virtual private networks (VPNs) in a process called **tunneling**. This process is used to allow a secure network to be implemented using a public network like the Internet. Imagine a company has just two offices, one in Australia and the other in Canada. When a computer in the Sydney office sends data to a computer in the Toronto office, the network packet is encrypted, put in the data portion of an Internet packet, and sent via the Internet to a special computer in the Toronto office. When it gets there, the computer at the Toronto office detects it is VPN data, extracts the encrypted data packet, decrypts it, and puts it onto its own local network where the target computer picks it up. As far as both

header				body
to address	from address	info	data length	data .....

FIGURE 14.9. Typical network packet format (simplified).

ethernet header			ethernet body			
			IP header			IP body
ethernet to addr.	ethernet from addr.	IP flag data len. etc.	IP to addr	IP from addr	other header IP info	Internet data .....

FIGURE 14.10. Internet IP packet inside Ethernet packet.

ends are concerned it looks as if both offices are on the same LAN and any data on the Internet is fully encrypted and secure.

Routing

If two computers are on the same piece of physical network, each can simply "listen" for packets that are destined for them—so sending messages between them is easy. If, however, messages need to be sent between distant machines—for example, if you are dialed into an ISP in the United Kingdom and are accessing a web server in the United States—the message cannot simply be broadcast to every machine on the Internet (Fig. 14.11). Instead, at each stage it needs to be passed in the right direction between different parts of the network. Routers perform this task. They look at the address of each packet and decide where to pass it. A local machine might need to simply put it onto the relevant local network, but if not it may need to pass it on to another intermediate machine.

Routers may be stand-alone boxes in network centers, or may be a normal computer. Often a file server acts as a router between a LAN and the global network.

As well as routers, networks are also linked by hubs and switches that make several different pieces of physical network behave as if they were one local network, and gateways that link different kinds of network. The details of these are not important, but they add more to the sheer complexity of even small networks.

Addresses

In order to send messages on the Internet or any other network, you need to have the address of where they are to go (or at least your computer needs it). In a phone network this is the telephone number and on the Internet it is an IP number. The *IP number* is a 32-bit number, normally represented as a group of four numbers between 0 and 255 (e.g. 212.35.74.132), which you will have probably seen at some stage when using a web browser or other Internet tool. It is these IP numbers that have are used by routers to send Internet data to the right places.

However, with any network there is a problem of getting to know the address. With phone numbers you simply look up the person's name in a telephone directory or by phoning the operator. Similarly, most networks have a naming scheme and some way to translate these into addresses. In the case of the

The 32-bit IP number space allows for 4 billion addresses. This sounds like quite a lot; however, these have been running out due to the explosive growth in the number of Internet devices and "wasted" IP numbers due to the way ranges of numbers get allocated to sub-networks. The new version of TCP/IP, IPv6, which is being deployed, has 128-bit addresses, which require 16 numbers (IPng, 2001). This allows sufficient unique IP addresses for every phone, PDA, Internet enabled domestic appliance, or even electronic paperclip.

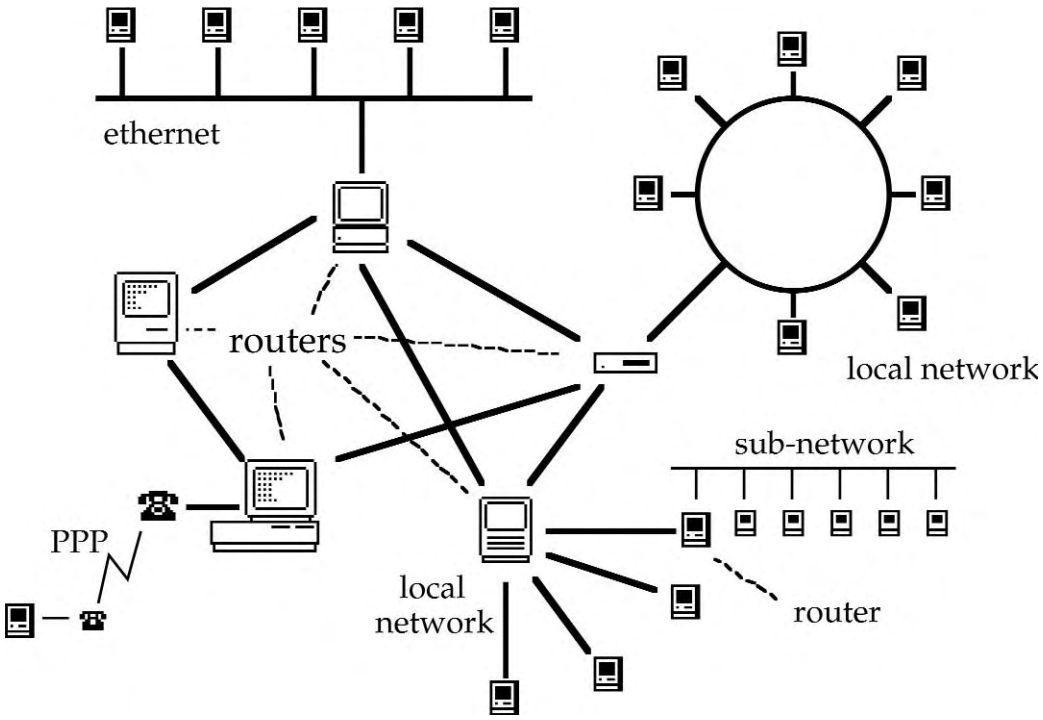


FIGURE 14.11. Routers send messages in the right direction through complex networks.

Internet, domain names (e.g., acm.org, www.hcibook.com, or magisoft.co.uk) are the naming system. There are many of these and they are changed relatively rapidly, so there is no equivalent of a telephone directory, but instead special computers called "domain name servers (DNS)" act as the equivalent of telephone directory inquiry operators. Every time an application needs to access a network resource using a domain name (e.g., to look up a URL) the computer has to ask a DNS what IP address corresponds to that domain name. Only then can it use the IP address to contact the target computer.

The DNS system is an example of a "white pages" system. You have an exact name and want to find the address for that name. In addition there are so-called "yellow pages" systems where you request, for example, a color postscript printer and are told of addresses of systems supplying the service. Sometimes this may be mediated by brokers who may attempt to find the closest matching resource (e.g., a nonpostscript color printer) or even perform translations (for example, the Java JINI framework; Edwards & Rodden, 2001).

This latter form of resource-discovery system is most important in mobile systems and ubiquitous computing where we are particularly interested in establishing connections with other geographically close devices.

A final piece in the puzzle is how one gets to know the address of the name server, directory service, or brokering service. In some types of network this may be managed by sending broadcast requests to a network ("Is there a name server out there?"). In the case of the Internet this is normally explicitly set for each machine as part of its network settings.

All Together . . .

If you are in your web browser and you try to access the URL http://www.meandeviation.com/qbb/, the following stages take place:

1. Send IP-level request to DNS asking for www.meandeviation.com
2. Wait for reply
3. DNS sends reply 64.39.13.108
4. Establish TCP level connection with 64.39.13.108
5. Send HTTP request ("GET/qbb/HTTP/1.1")
6. Web server sends the page back in reply
7. Close TCP connection

Most of these stages are themselves simplified, all will involve layering on top of lower-level networks, and most stages involve several sub-stages (e.g., establishing a TCP level connection requires several IP-level messages).

The basic message is that network internals are multilevel, multistage, and pretty complicated.

Decentralizing: Peer–Peer and Ad-Hoc Networks

Traditional Internet and web applications tended to work through client–server paradigms in which a user's client application accesses a well-known application such as a web server. However, there has been a growth of more decentralized models at both higher and lower protocol levels (Androutsellis-Theotokis & Spinellis, 2004).

At an application level, peer–peer file sharing works by having clients running on users' own PCs talk directly to one another broadcasting requests for particular files by name or type to all connected clients. These then forward the requests to their connected nodes until some maximum hop count is hit. Often semi-centralized services are used to establish initial connections, but thereafter everything is done at the "edges" of the network. These protocols and applications have some technical strengths but their origins are rooted in legal disputes on the distribution of copyright material. Recent research has also shown that the anonymity of peer–peer file sharing also allows "deviant" subgroups sharing illicit material (Hughes, Gibson, Walkerdine, & Coulson, 2006).

At a lower level, ad-hoc networks build more structured higher-level networks for computers that have point-to-point connections, often through wireless connectivity. This can be used to set up a network in a meeting, or at a larger level could even allow everyone in a pop festival to have Internet access through a small number of wireless hot spots. In an ad hoc network, the low-level software works out what machines are connected where and routes messages from end to end even if the two end-point computers do not have direct access. The difficult thing in an ad hoc network is not just setting up this routing information, but the dynamics: dealing with the fact that machines constantly move around, perhaps going out of range of one another, may be turned off, or may join the network late.

Network Management

Those most obviously exposed to this complexity are the engineers managing large national and international networks, both data networks such as the main Internet backbones, and telecoms networks. The technical issues outlined above are compounded by the fact that the different levels of network hardware and network management software are typically supplied by different manufacturers. Furthermore, parts of the network may be owned and managed by a third party and shared with other networks. For example, a transatlantic fiber optic cable may carry telecoms and data traffic from many different carriers.

When a fault occurs in such a network it is hard to know whether it is a software fault or a hardware fault, where it is happening, or who is responsible. If you send engineers out to the wrong location it will cost both their time and the time the service is unavailable. Typically the penalties for inoperative or reduced quality services are high; you need to get it right, fast.

This is a specialized and complex area, but clearly of increasing importance. It poses many fascinating UI challenges: how to visualize complex multilayered structures and how to help operators trace faults in these. Although I know that it is a topic being addressed by individual telecoms companies, published material in the HCI literature is minimal. The exception is visualization of networks, both physical and logical, which is quite extensive (Dodge and Kitchin's Atlas of Cyberspace and associated website (http://www.cybergeography.org/atlas/) is the comprehensive text in this area; Dodge & Kitchin, 2001).

Ordinary systems administrators in organizations face similar problems albeit on a smaller scale. A near-universal experience and consequently of misconfigured e-mail systems, continual network failures and performance problems certainly suggests this is an area ripe for effective interface solutions, but again there is very little in the current HCI literature.

Finally, it appears that everyone is now a network administrator; even a first-time home PC user must manage modem settings, name server addresses, SMTP and POP servers, and more. It is interesting that the interface used by most such users is identical to that supplied for full systems administrators. Arguably this may ease the path for those who graduate from single machines to administering an office or organization (perhaps less than 5% of users). Unfortunately, it makes life intolerable for the other 95%! The only thing that makes this possible at all is that the "welcome" disks from many ISPs and the Wizards shipping with the OS offer step-by-step instructions or may automatically configure the system. These complications are compounded if the user wishes to allow access through more than one ISP, or connect into a fixed network. As many home users now have several PCs and other devices that need to be networked, this is not a minor issue.

In the first edition of this handbook I wrote, "If we look at the current state of the two most popular PC systems, Microsoft Windows and Mac OS, the picture is not rosy.". Sadly things have not improved dramatically; you only have to watch a room full of computer scientists at a meeting trying to sort out wireless connections.

On Windows, if you do want to change your network settings you need to set several IP numbers, web proxies, and SMTP settings in several control panels. In fact as I sat here now writing I tried and failed to find the place to change the web proxy setting on my PC. It is possible to set up alternate settings for networks if you move between two locations, but more than that and you are lost! On MacOS the situation has improved in the last few years and now nearly all settings are in one place under "Network Settings." Changing locations with a laptop is simply a matter of selecting "Location . . ." from the main menu. Perversely, the default printer that used to be attached to your location on the older MacOS 9 interface now has to be changed by hand each time!

The problems in both emphasize partly the intrinsic complexity of networking—yes, it does involve multiple logically distinct settings, many of which relate to low-level details. However, it also exposes the apparent view that those involved in network administration are experts who understand the meaning of various internal networking terms. This is not the case even for most office networks, and certainly not at home.

And this is just initially setting up the system to use. For the home user, debugging faults has many of the same problems as large networks. You try to visit a website and get an error box. Is the website down? Are there problems with the wireless LAN, the broadband modem, the phone line, or the ISP's hardware? Are all your configurations settings right? Has a thunderstorm 3,000 miles away knocked out a vital network connection? Trying to understand a multilayered, non-localized and, when things work, largely hidden system is intrinsically difficult and where diagnostic tools for this are provided, they assume an even greater degree of expertise.

The much-heralded promise of devices that connect to one another within our homes and about our bodies is going to throw up many of the same problems. Some old ones may ease as explicit configuration becomes automated by self-discovery between components, but this adds further to the "hiddenness" and thus difficulty in managing faults, security, and so on. You can imagine the scenario: The sound on my portable DVD stops working and produces a continuous noise—why? There are no cables to check of course (wireless networking), but hours of checking and randomly turning devices on and off narrows the problem down to a fault in the washing machine which is sending continuous "I finished the clothes" alerts to all devices in the vicinity.

Network Awareness

One of the problems noted above is the hiddenness of networks. This causes problems when things go wrong as one hasn't an appropriate model of what is going on, but also sometimes even when things are working properly.

We discussed earlier some of the network properties that may affect usability: bandwidth, delay, jitter, and so forth. These are all affected to some extent by other network loads, the quality of current network connections, and so on. So predicting performance (and knowing whether to panic if things appear to go slow) needs some awareness of the current state of the network.

Personal computers using wireless networks usually offer some indication of signal strength (if one knows where to look and what it means) although this is less common for line quality for modems. As wireless devices and sensors become smaller they will not have suitable displays for this and explicitly making users aware of the low-level signal strength of an intelligent paperclip may not be appropriate. However, as interface designers we do need to think how users will be able to cope and problem-solve in such networks.

Only more sophisticated network management software allows one to probe the current load on a network. This does matter. Consider a small home network with several PCs connected through a single modem. If one person starts a large download they will be aware that this will affect the performance of the rest of their web browsing. Other members of the household will just experience a slowing down of everything and not understand why. If the cause is a file-sharing utility such as Gnutella, even the person whose computer is running it may not realize its network impact. In experiments at MIT the level of network traffic has been used to "jiggle" a string hanging from the ceiling so that heavy traffic leads to a lot of movement (Wisneski et al., 1998). The movements are not intrusive so give a general background awareness of network activity. Although supplying a ceiling-mounted string may not be the ideal solution for every home, other more prosaic interface features are possible.

Cost awareness is also very important. In the United Kingdom, first generation GSM mobile data services are charged by connection time. So knowing how long you have been connected and how long things are likely to take becomes critical. If these charges differ at peak hours of the day, calculating whether to read your e-mail now, or do a quick check and download the

big attachments later can become a complex decision. The move to data-volume-based charging means the user needs to estimate the volume of data that is likely to be involved in initiating an action versus the value of that data—do you want to click on that link if the page it links to includes large graphics, perhaps an applet or two? Terrestrial broadband networks have largely shifted to fixed monthly fees and this has dramatically changed the way people perceive the Internet, from smash-and-grab interactions to surf-and-play.

Network Confusion

If the preceding doesn't sound confusing enough, the multilayered nature of networked applications means that it is hard to predict the possible patterns of interference between things implemented at different levels or even at the same level. Again this is often most obvious when things go wrong, but also because unforeseen interactions may mean that two features that work perfectly well in isolation may fail when used together.

This problem, feature interaction, has been studied particularly in standard telecoms (although it is certainly not confined to them). Let's look at an example of feature interaction. Telephone systems universally apply the principle that the caller, who has control over whether and when the call is made, is the person who pays. In the exceptions (free-phone numbers, reverse charges) special efforts are made to ensure that subscribers understand the costs involved. Some telephone systems also have a feature whereby a caller who encounters a busy line can request a callback when the line becomes free. Unfortunately at least one company implemented this callback feature so that the charging system saw the callback as originating from the person who had originally been called. Each feature seemed to be clear on its own, but together meant you could be charged for calls you didn't want to make.

With N features there are $N(N-1)/2$ possible pairs of interactions to consider, $N(N-1)(N-2)/6$ triples, and so on. This is a well-recognized (but not solved) problem with considerable efforts being made using, for example, formal analysis of interactions to automatically detect potential problems. It is worth noting that this is not simply a technical issue; the charging example shows that it is not just who pays that matters, but also the perceptions of who pays. This particular interaction would have been less of a problem if the interface of the phone system had, for example, said (in generated speech), "You have had a call from XXX; press 'callback' and you will be charged for this call." The hybrid-feature interaction research group at Glasgow are attempting to build a comprehensive list of such problems in telecoms (HFIG, 2001), but these sorts of problems are likely to be found increasingly in related areas such as ubiquitous computing and resource discovery.

Exploiting the Limitations: Seamfulness and Virtual Locality

Some companies cynically exploit this user confusion over network charging and in the United Kingdom there have been some high-profile news stories about teenagers running up thousands of pounds of debt after innocently signing up to ringtone delivery services.

Happily there are also more positive uses of the limitations of networks. Given suitable awareness mechanisms for network strength and connectivity, people are very resourceful in exploitation. You will have experienced the way mobile-phone users get to know the sweet spots for their networks when in areas of poor coverage, learning to get out of doors, away from big buildings, or up hills. This can get sophisticated. One mixed-reality experiment, part of the Equator project, consisted of an outside game where real players running in the street were pitted against virtual characters manipulated remotely. The "real" players learned to hide in the GPS shadow of buildings so that their location could not be detected by the virtual participants (Benford et al., 2003). Similar effects have been seen in WiFi-based games where participants get to know the regions of good and bad coverage and may use these to seek out or avoid remote interactions.

In these ways, users do more than cope; they actively exploit the limitations of networks and sensing. The Pirates game did this with RF technology. RF beacons represent islands in an ocean, the locality of the RF signals corresponding to the limited land area of the island (Björk, Falk, Hansson, & Ljungstrand, 2001). This notion of exploiting network limitations has led to the idea of *seamful games*, deliberately designed so that variations in WiFi coverage and connectivity are part of the game play (Chalmers et al., 2005).

Even more strange is a recent game "Hitchers." In this you pick up and drop off hitchers with your phone, just as you might pick up a hitchhiker on the road. You can only pick up a hitcher in the same mobile cell as the last person who dropped the hitcher off. The hitchers are actually stored and managed centrally, but the effect is as if the hitchers were only accessible in a small region. Here, even though there is no real limitation of access, a limitation is constructed to make a more interesting game: *virtual locality*. You can imagine virtual locality being used in synchronous applications—for example, allowing a phone user to broadcast to people in their vicinity apparently as if it were a limited range transmission, whereas in reality it is centrally managed, based on GPS or phone cell location?

The Network Within

So far the story is pretty bleak from a user interface viewpoint: a complex problem, of rapidly growing importance, with relatively little published work in many areas. One good thing as a HCI practitioner about understanding the complexity of networks is that they help us understand better the workings of the human cognitive and motor system.

For at least five decades, computational models have been used to inspire cognitive models. Also of course cognitive and neurological models have been used to inspire computational models in artificial intelligence and neural networks. However, our bodies are not like a single computer, but in various ways more like a networked system.

This is because, first, several things can happen at once. The interacting cognitive subsystems (ICS) model from APU Cambridge (Barnard & May, 1995) takes this into account by looking

at various parts of the cognitive system, the conversions between representations between these parts, and the conflicts that arise if the same part is used to perform different tasks simultaneously. Similarly, the very successful PERT-style GOMS analysis used on the NYNEX telephone operators' interface used the fact that the operator could be doing several things simultaneously with no interfering parts of their bodies and brains (John, 1990; Gray, John, & Atwood, 1992).

We are also like a networked system in that signals take an appreciable time to get from our senses to our brains and from our brains to our muscles. The famous homunculus from Card, Moran, and Newell's (1983) Model Human Processor makes this very clear with timings attached to various paths and types of mental processing. In fact, the sorts of delays within our bodies (from 50–200 ms on different paths) are very similar to those found on international networks.

In industrial control one distinguishes between open-loop and closed-loop control systems. *Open-loop control* is where you give the machine an instruction and assume it does it correctly (like a treasure map: "Ten steps North, turn left, three steps forward, and dig"). This assumes the machine is well calibrated and predictable. In contrast, *closed-loop control* uses sensors to constantly feedback and modify future actions based on the outcomes of previous ones (e.g., "Follow the yellow brick road until you come to the Emerald City"). Closed-loop control systems tend to be far more robust, especially in uncontrolled environments, like the real world.

Not surprisingly our bodies are full of closed-loop control systems (e.g., the level of carbon dioxide in your lungs triggers the breathing reflex). However, closed-loop control can become difficult if there are delays (you have not received feedback from the previous action when starting the next one). Delays either mean one has to slow down the task or use some level of prediction to work out what to do next, based on feedback of actions before the last one. This breakdown of closed-loop control in the face of (especially unexpected) delays is one of the reasons hand-eye coordination tasks, such as mouse movement, break down if delays exceed a couple of hundred milliseconds. The feedback loops in our bodies for these tasks assume normal delays of around 200 ms and are robust to variations around this figure, but adding delays beyond this starts to cause breakdown.

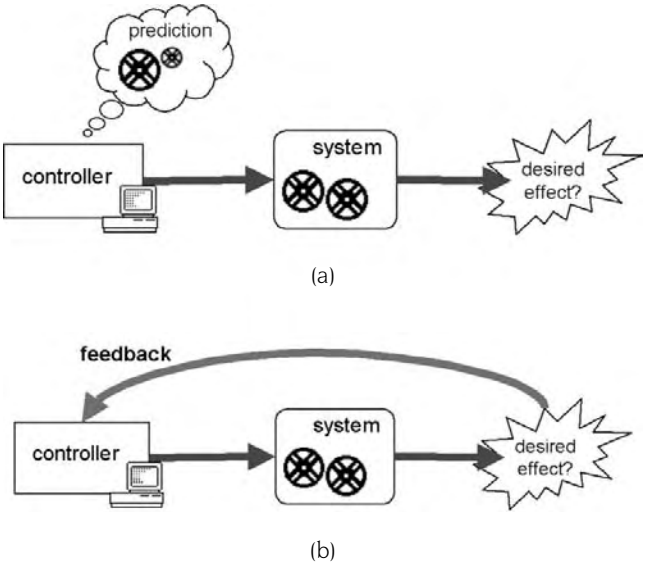


FIGURE 14.12. (a) Open-loop control. (b) Closed-loop control.

The delays inside our bodies cause other problems too. The path from our visual cortex into our brain is far faster (by 100 ms or so) than that from our touch and muscle tension sensors around our bodies. If we were designing a computer system to use this information, we might consider having a short 100 ms tape loop, so that we could store the video input until we had the appropriate information from all senses. However, the sheer volume of visual information means that our brains do not attempt to do this. Instead there is a part of your brain that predicts where it "thinks" your body is and what it is feeling based on previous nerve feedback and what it knows the muscles have been asked to do. The same bit of the brain then monitors what actually did happen (when the nerve signals have made their way up the spinal column to the brain) and gives an uncomfortable or shocked sensation when a mismatch occurs—for example, if you go to pick something up, but because of poor light or a strange-shaped object you touch it earlier or later than you would expect. Tickling is also connected with this lack of ability to predict the sensations (this is why it is difficult to tickle yourself).

Race conditions also occur within this networked system of our bodies—for example, getting letter inversions whilst typing where signals to the two hands get processed in the wrong order. Brewster and Dix (1994) also used race conditions to understand what goes wrong in certain kinds of mis-hits of on-screen buttons. In certain circumstances two almost simultaneous "commands" from our brains to our hands to release the mouse button and to our arms to move to a new mouse location can get out of order, meaning the mouse moves out of the target before it is released. This analysis allowed us to design an experiment that forced this very infrequent error to occur much more frequently and therefore make it easier to assess potential solutions.

NETWORKS AS PLATFORMS

Algorithms and Architectures for Distributed Interfaces

User interfaces are hard enough to construct on a single machine; concurrent access by users on networked machines is a nightmare!

Happily, appropriate algorithms, architectures, toolkits and frameworks can help—a bit.

Accessing Shared Objects

We saw earlier how race conditions within networked systems can lead to inconsistencies within the user interface and within the underlying data structures. Fortunately, there are a range of techniques for dealing with this.

Locking

The standard technique, used in databases and file systems, for dealing with multiple accesses to the same data object is locking. When a user's application wants to update a particular database record, it asks the database manager for a lock on the record. It can then get a copy of the record and send back the update, knowing that nothing else can access it in the meantime. Users are typically unaware that locking is being performed—the act of opening a file or opening an edit form for a database record establishes the lock and later, when the file or the edit form is completed and closed, the lock is released.

Although this is acceptable for more structured domains there are problems in more dynamic domains such as shared editing. Locking a file when one user is editing it is no good as we want several people to edit the same file at the same time. In these cases more lightweight forms of locking can be used at finer granularities: at paragraph, sentence, or even per-character level. For example, the act of clicking over a paragraph to set a text-entry position may implicitly request a paragraph lock, which is released when you go on to edit another paragraph. However, implicit and informal locks, because they are not apparent, can lead to new problems. For example, a user may click on a paragraph and do some changes, but before moving on to another part of the document, get interrupted. No one else can then edit the paragraph. To avoid this, the more informal locks are often time-limited or can be forcibly broken by the server if another user requests a lock on the same object.

Replication

In collaboration systems such as Lotus Notes/Domino or source code systems such as CVS, users do not lock central copies of data, but instead each user (or possibly each site) has their own replica of the complete Notes database/CVS tree. Periodically these replicas are synchronized with central copies or with each other. Updates can happen anywhere by anyone with a replica. Conflicts may of course arise if two people edit the same note between synchronizations. Instead of preventing such conflicts, the system (and software written using it) accepts that such conflicts will occur. When the replicas synchronizes, conflicts are detected and various (configurable) actions may occur: flagging the conflicts to users, adding conflicting copies as versions, and so on.

This view of "replicate and worry later" is essential in many mobile applications, as attempts to lock a file whilst disconnected would first of all require waiting until a network connection could be made, and, worse, if the network connection is lost whilst the lock is still in operation, could lead to files being locked for very long periods. Other examples of replication in research environments include the CODA at CMU, which allows replication of a standard UNIX file system (Kistler & Satyanarayanan, 1992) and Liveware, a contact-information system that replicates and synchronizes when people meet in a manner modeled after the spread of computer viruses (Witten, Thimbleby, Coulouris, & Greenberg, 1991). Although Liveware is now quite a few years old and was "spread" using synchronizing floppy disks, the same principle is now being suggested for PDAs and other devices to exchange information via IrDA or Bluetooth.

Optimistic Concurrency for Synchronous Editing

The "do it now and see if there are conflicts later" approach is called "optimistic concurrency," especially in more synchronous settings. For example, in a shared editing system the likelihood of two users editing the same sentence at the same time is very low. An optimistic algorithm doesn't bother to lock or otherwise check things when the users start to edit in an area, but in the midst or at the end of their edits checks to see if there are any conflicts, and attempts to fix them.

There are three main types of data that may be shared:

- **Orthogonal data**—where the data consists of attributes of individual objects/records that can all be independently updated
- **Sequential data**—particularly text, but any form of list where the order is not determined by an attribute property
- **Complex structural data**—such as directory trees, taxonomic categories, and so on.

In terms of complexity for shared data, these are in increasing difficulty.

Orthogonal data, although by no means trivial, is the simplest case. There is quite a literature on shared graphical editors, which all have this model: independent shapes and objects with independent attributes such as color, size, or position. When merging updates from two users, all one has to do is look at each attribute in turn and see whether it has been changed by only one user, in which case the updated value is used, or if it has been changed by both, in which case either the last update is used or the conflict is flagged.

Structured data is most complicated. What do you do if someone has created a new file in directory D, but at the same time someone else has deleted the directory? I know of no optimistic algorithms for dealing effectively with this in the CSCW literature. CODA deals with directory structures (normal UNIX file system) but takes a very simple view of this as it only flags inconsistencies and doesn't attempt to fix them.

Algorithms for shared text editing sit somewhere between the two and have two slightly different problems, both relating to race conditions when two or more users are updating the same text:

- **Dynamic pointers**—if user A is updating an area of text in front of user B, then the text user B is editing will effectively move in the document.
- **Deep conflict**—what happens if user A and user B's cursors are at the very same location and they perform insertions/deletions

Figure 14.13 shows an example of the first of these problems. The deeper conflict occurs when both cursors are at the same point, say after the "Y" in "XYZ." Adonis types "A" and at the same time Beatrice types "B"—should we have "XYABZ" or "XYBAZ," or perhaps even lose one or other character? Or if Adonis types "A" and Beatrice presses "delete," should we have "XYZ" or "XAZ"?

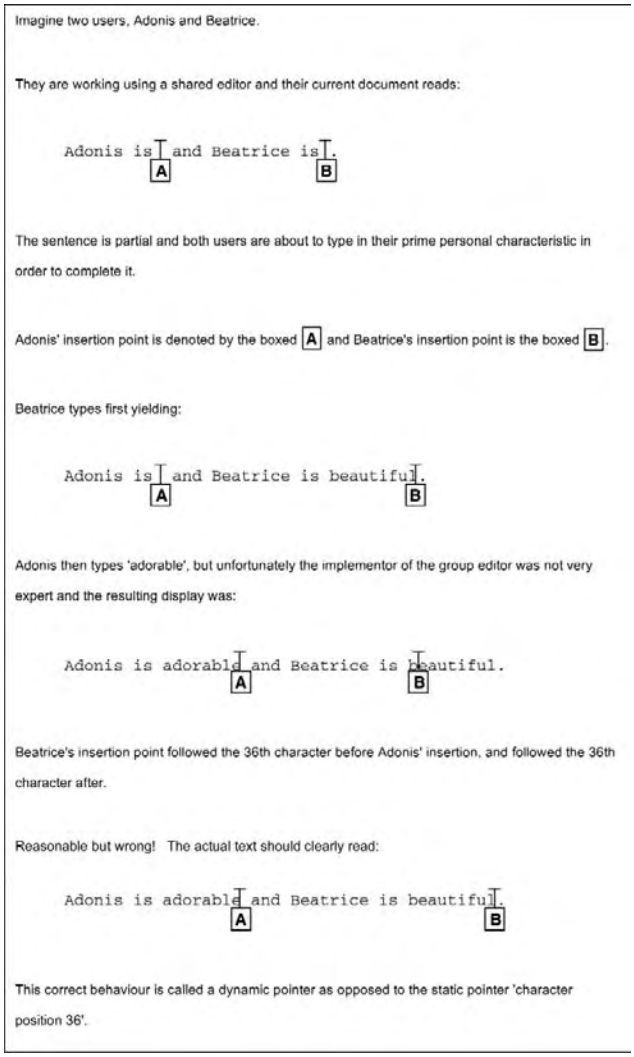


FIGURE 14.13. Dynamic pointers from Dix (1995).

A number of algorithms exist for dealing with this (Sun & Ellis, 1998; Mauve, 2000; Vidot, Cart, Ferriz, & Suleiman, 2000), including a retrofit to Microsoft Word called CoWord (Xia, Sun, Sun, Chen, & Shen, 2004). Most of these stem from the dOPT algorithm used in the Grove editor (Ellis & Gibbs, 1989). These algorithms work by having various transformations that allow you to reorder operations. For example, if we have two insertions (labeled "a" and "b") performed at the same time at different locations

a) insert texta at location n
b) insert textb at location m

but decide to give insert a preference, then we have to transform b to b′ as follows:

b′) if (m < n) insert textb at location m (i)
if (m = n) insert textb at location m (ii)
if (m > n) insert textb at location m+length(texta) (iii)

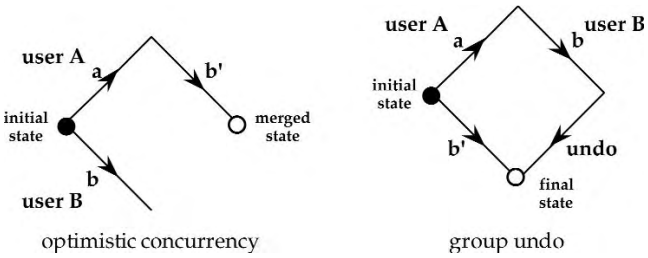


FIGURE 14.14. Multi-user transformations.

Case (i) says that if the location of insert "b" is before insert "a" you don't have to worry. Case (iii) says that if it is after insert "a" you have to shift your insert along accordingly. Case (ii) is the difficult one where a conflict occurs and has to be dealt with carefully to ensure that the algorithm generates the same results no matter where it is. The version above would mean that B's cursor gets left behind by A's edit. The alternative would be to make case (ii) the same as case (iii) which would mean B's cursor would be pushed ahead of A's.

In early work in this area I proposed regarding dynamic pointers as first-class objects and using these in all representations of actions (Dix, 1991, 1995). This means that rules like case (i) and case (iii) happen "for free," but the deep conflict case still needs to be dealt with specially.

Groupware Undo

The reason that undo is complicated in groupware is similar to the problems of race conditions in optimistic concurrency.

In the case of optimistic concurrency user A has performed action **a** and user B has performed action **b**, both on the same initial state. The problem is to transform user B's action into one **b′** that can be applied to the state after action **a** yet still mean the "same things" as the original action **a**.

In the case of groupware undo we may have the situation where user A has performed action **a**, followed by user B performing action **b**, then user A decides to undo action **a**. How do we transform action **b** so that the transformed **b′** means the same before action **a** as **b** did after?

Similar, but slightly different transformation rules can be produced for the case of undo and dynamic pointers can be used for most cases.

As with optimistic concurrency there is slightly more work on group undo in shared graphical editors where the orthogonal data makes conflicts easier (Berlage & Spenke, 1992).

Real Solutions?

Although these various algorithms can ensure there is no internal inconsistency and that all participants see the same thing, they do not necessarily solve all problems. Look again at the case of Alison and Brian's chat earlier in the chapter. Certainly in the case of group undo, when Abowd and Dix (1992) published the

first paper on the topic, we proposed various solutions, but also recommended that as well as an explicit undo button, systems ought to provide sufficient history to allow users to recreate what they want to without using the undo button. This is not because it is impossible to find a reasonable meaning for the undo button, but because in the case of group undo, there are several reasonable meanings. Choosing the meaning a user intends is impossible and so it may sometimes be better not to guess.

Architectures for Networked Systems

Software architecture is about choosing what, in terms of code and functionality, goes where. For applications on a single machine these are often logical distinctions between parts of the code. For networked systems, "where" includes physical location, and the choice of location makes an enormous difference to the responsiveness of the system.

The simplest systems are usually centralized client–server architectures in which the majority of computation and data storage happens in the central server. Many of the problems of race conditions and potential inconsistencies disappear. However, this means that every interaction requires a network interaction with the server meaning that feedback may be very slow. The opposite extreme are replicated peer–peer architectures where all the code is running on users' own PCs, and the PCs communicate directly with one another. Feedback can now be instantaneous, but the complexity of algorithms to maintain consistency, catch-up late joiners, and so on, can be very complex. Most systems operate somewhere between these two extremes with a central "golden copy" of shared data, but with some portion of the data on individual PCs, PDAs, or other devices in order to allow rapid feedback. A notable exception to this is peer–peer sharing networks such as Gnutella where the only central resource is a sort of switchboard to allow client programs to contact one another. The reason for this is largely legal as these are often used to share copyright media! One reason this works is that the data is static; the song "Yesterday" is the same now as when it was first recorded in 1965—although given it has the greatest number of covers of any song this is perhaps not the best example!

In web applications options are constrained by the features allowed in HTML and web browsers. Note that even a web form is allowing some local interaction (filling in the form) as well as some centralized interaction (submitting it). Applets allow more interaction, but the security limitations mean that they can only talk back to the server where they originated. Thus true peer–peer architectures are impossible on the web, but can be emulated by chat servers and similar programs that relay messages between clients. For Intranets it is easier to configure browsers so that they accept applets as trusted and thus with greater network privileges, or to include special plug-in components to perform more complicated actions.

Mobile systems have yet more issues as they need to be capable of managing disconnected operation (when they have no physical or wireless connection to the network). This certainly means keeping cached copies of central data. For example, the AvantGo browser for PDAs downloads copies of web pages onto a PDA and synchronizes these with their latest versions every time the PDA is docked into a desktop PC (AvantGo, 2001). Although this is now beginning to change with higher-bandwidth mobile-data services, it is still the case that access whilst mobile is slower and more expensive than whilst using a fixed connection. This pushes one towards replicated architectures with major resynchronization when connected via cheap/fast connections and more on-demand or priority-driven synchronization when on the move.

Supporting Infrastructure

In order to help manage networked applications various types of supporting infrastructure are being developed.

Awareness Servers

These keep track of which users are accessing particular resources (e.g., visiting a particular web page) so that you can be kept informed as to whether others are "near" you in virtual space, or whether friends are online and active. (Palfreyman & Rodden, 1996; ICQ, 2001; SUN Microsystems, 2001)

Notification Servers

These serve a similar role for data, allowing client programs to register an interest in particular pieces of shared data. When the data is modified or accessed the interested parties are "notified." For example, you may be told when a web page has changed or when a new item has been added to a bulletin board. Some notification servers also manage the shared data (Patterson, Day, & Kucan, 1996) whilst others are "pure," just managing the job of notification (Ramduny, Dix, & Rodden, 1998).

Event/Messaging Systems

These allow different objects in a networked environment to send messages to one another in ways which are more convenient than those allowed by the raw network. For example, they may allow messages to be sent to objects based on location-independent names, so that objects don't have to know where each other are.

Resource Discovery

Systems such as the Java JINI framework and Universal Plug-and-Play allow devices to find out about other devices close to them—for example, the local printer—and configure themselves to work with one another. As ubiquitous and mobile devices multiply, this will become increasingly important.

Web Service and Remote Procedure Call

Web service frameworks such as SOAP or XML-RPC, or non-web remote procedure mechanisms such as Java's RMI or CORBA, allow applications on different machines to easily connect to

one another even though they may be written in different programming languages and run on different operating systems. In all cases the parameters or arguments have to be serialized into a binary or ASCII test stream to be sent. As Fig. 14.15 shows these are designed to be read by machines, not people!

HISTORY, FUTURES, AND PARADIGM SHIFTS

History

Given the anarchic image of the web, it is strange that the Internet began its development as a United States military project. The suitability of the Internet for distributed management and independent growth stems not from an egalitarian or anti-centralist political agenda, but from the need to make the network resilient to nuclear attack, with no single point of failure.

In the 1970s when the Internet was first developing, it and other networks were mainly targeted at connecting together large computers. It was during the 1980s, with the rise of personal computing, that local networks began to become popular.

However, even before that point, very local networks at the lab-bench level had been developed to link laboratory equipment—for example, the IEEE488 designed originally to link Hewlett Packard's proprietary equipment and then becoming an international standard. Ethernet too began life in commercial development at Xerox before becoming the de facto standard for local networking.

Timeline—Key Events for the Internet

1968	First proposal for ARPANET –military and government research contracted to Bolt, Beranek, and Newman
1971	ARPANET enters regular use
1973/4	Redesign of lower level protocols leads to TCP/IP
1983	Berkeley TCP/IP implementation for 4.2BSD—public domain code
1980s	Rapid growth of NSFNET –broad academic use
1990s	WWW and widespread public access to the Internet
2000	WAP on mobile phones Web transcends the Internet

Although it is technical features of the Internet (decentralized, resilient to failures, hardware independent) that have made it possible for it to grow, it is the web that has made it become part of popular consciousness. Just as strange as the Internet's metamorphosis from military standard to anarchic cult, is the web's development from medium of exchange of large high-energy physics datasets to e-commerce, home of alternative web pages and online sex!

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns1:sayHelloTo
        xmlns:ns1="Lookup"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <name xsi:type="xsd:string">Hello World</name>
    </ns1:sayHelloTo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FIGURE 14.15. SOAP XML encoding of Lookup ("Hello World").

Paradigm Shift

During the 1970s through to the mid-1990s, networks were a technical phenomenon, enabling many aspects of business and academic life, but with very little public impact. However, this has changed dramatically over the last five years and now we think of a networked society. The Internet and other network technologies, such as SMS text messages, are not only transforming society but, at a popular and cultural level, defining an era.

International transport, telecommunications, and broadcasting had long before given rise to the term *global village*. However, it seems this was more a phrase waiting for a meaning. Until recently the global village was either parochial (telephoning those you already know) or sanitized (views of distant cultures through the eyes of the travel agent or television camera). It is only now that we see chat rooms and web homepages allowing new contacts and friendship around the world or at least amongst those affluent enough to have Internet access.

Markets too have changed due to global networks. It is not only the transnationals who can trade across the world, and even my father-in-law runs a thriving business selling antiques through eBay.

Marketing has also had to face a different form of cross-cultural issue. Although selling the same product, a hoarding in Karachi may well be different from an advert in a magazine in Kentucky, reflecting the different cultural concerns. Global availability of web pages changes all that. You have to create a message that appeals to all cultures, a tall order. Those who try to replicate the targeting of traditional media by having several country-specific websites may face new problems; the global access to even these country-specific pages means that the residents of Kentucky and Karachi can compare the different adverts prepared for them, and in so doing see how the company views them and their cultures.

Economics drives much of popular, as well as business, development of networked society. One of the most significant changes in the United Kingdom in recent years was due to changes in charging models. In the United States, local calls have long been free and hence so were Internet connections to local points of presence. The costs of Internet access in the United Kingdom (and even more important, the perception of the cost) held back widespread use. The rise of free or fixed-charge unmetered access changed nearly overnight the acceptability and style of use. Internet access used to be like a lightning guerrilla attack into the web territory, quick in and out before the costs mounted up, but is now a full-scale occupation.

The need for telecoms companies across the world to recover large investments in wireless-band franchises combined with use rather than connection-based charging made possible by GPRS and third-generation mobile services, make it likely that we will see a similar growth in mobile access to global networked information and services.

In an article in 1998, I used the term *PopuNET* to refer to a change in society that was not yet there, but would come. PopuNET is characterized by network access

everywhere, everywhen by everyone

This pervasive, permanent, popular access is similar to the so-called "Martini principle" applied more recently to mobile networking—anytime, anyplace, anywhere. Of course, Martini never pretends to be anything but exclusive, so not surprisingly these differ on the popular dimension! "Anyplace, anywhere" does correspond to the pervasive "everywhere" and "anytime" to the permanent "everywhen." However, there is a subtle difference, especially between "anytime" and "everywhen.". "Anytime" means that at any time you choose you can connect. "Everywhen" means that at all places and all times you are connected. When this happens, one ceases to think of "connectedness" and it simply becomes part of the backdrop of life.

The changes in charging models have brought the United Kingdom closer to "everywhen" and the United States and many other parts of the world are already well down the path. Always-on mobile connectivity will reinforce these changes.

PopuNET will demand new interfaces and products, not just putting web pages on TV screens or spreadsheets on fridge doors. What these new interfaces will be is still uncertain.

Futures (Near)

It is dangerous to predict far into the future in an area as volatile as this. One development that is already underway, which will make a major impact on user interfaces, is short-range networking which will enable various forms of wearable and ubiquitous networks. Another is the introduction of network appliances, which will make the home "alive" in the network.

We have considered network aspects of continuous media at length. The fact that the existing Internet TCP/IP protocols do not enable guaranteed quality-of-service will put severe limits on its ability to act as an infrastructure for services such as video-on-demand or video-sharing between homes. The update to TCP/IP which has been under development for several years, IPv6, will allow prioritized traffic, but it falls short of real guaranteed QoS (IPng, 2001).

It seems that this is impasse. One of the reasons that IPv6 has taken so long is not the technical difficulty, but backwards compatibility and the problems of uptake on the existing worldwide infrastructure, so evolutionary change is hard. However, revolutionary change is also hard; one cannot easily establish a new parallel international infrastructure overnight. Or perhaps one can.

Mobile phone services have started with an infrastructure designed for continuous voice and are, through a series of quite dramatic changes, moving this towards a fully mixed media/data service—and it is a global network. Furthermore, more and more non-web-based Internet services are using HTTP, the web protocol, to talk to one another in order to be "firewall friendly." This means that mobile phones and PDAs can be web-connected without being Internet-connected. So perhaps the future is not an evolution of Internet protocols, but a gradual replacement of the Internet by nth-generation mobile networks. Instead of the Internet on phones, we may even see mobile phone networking standards being used over wires.

References

All web links below and other related links and material at: http://www.hiraeth.com/alan/hbhci/network/

Abowd, G. D., & Dix, A. J. (1992). Giving undo attention. *Interacting with computers*, *4*(3), 317–342.

AlexaWSP (2006). *Alexa Web Search Platform Service*. Retrieved March 13, 2006, from http://pages.alexa.com/prod_serv/web_search_platform.html

Androutsellis-Theotokis, S., & Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys, 36*(4), 335–371.

Arehart, C., Chidambaram, N., Guruprasad, S., Homer, A., Howell, R., Kasippillai, S., et al. (2000). *Professional WAP*. Birmingham, UK: Wrox Press.

Arons, B. (1997). SpeechSkimmer: A system for interactively skimming recorded speech. *ACM Transactions on Computer–Human Interaction (TOCHI), 4*(1), 3–38.

AvantGo (2001). *AvantGo, Inc.* Retrieved March 13, 2006 from http://www.avantgo.com/

Barnard, P., & May, J. (1995). Interactions with advanced graphical interfaces and the deployment of latent human knowledge. In F. Paternò (Ed.), *Eurographics Workshop on the Design, Specification and Verification of Interactive Systems*. Berlin: Springer Verlag, 15–49.

Bellotti, V. (1993). Design for privacy in ubiquitous computing environments. *Proceedings of CSCW'93* (pp. 77–92). New York: ACM Press.

Benford, S., Anastasi, R., Flintham, M., Drozd, A., Crabtree, A., & Greenhalgh, C., et al. (2003). Coping with uncertainty in a location-based game. *IEEE Pervasive Computing, 2*(3), 34–41.

Benford, S., Bowers, J., Fahlen, L., Mariani, J., & Rodden, T. (1994). Supporting cooperative work in virtual environments. *The Computer Journal, 37*(8), 635–668.

Berlage, T., & Spenke, M. (1992). The GINA interaction recorder. *Proceedings of the IFIP WG2.7 Working Conference on Engineering for Human–Computer Interaction*, Amsterdam: North Holland, Ellivuori, Finland, 69–80.

Björk, S., Falk, J., Hansson, R., & Ljungstrand, P. (2001). Pirates!—Using the physical world as a game board. In M. Hirose (Ed.), *Proceedings of Interact 2001* (pp. 9–13). Amsterdam: IOS Press.

Bluetooth. (2001). *Official bluetooth SIG website*. Retrieved March 13, 2006, from http://www.bluetooth.com/

Buxton, W., & Moran, T. (1990). EuroPARC's integrated interactive intermedia facility (IIIF): Early Experiences. In S. Gibbs & A. A. Verrijn-Stuart (Eds.), *Multi-user interfaces and applications, Proceedings of IFIP WG8.4 Conference*, (pp. 11–34). Heraklion, Greece. Amsterdam: North-Holland.

Campbell, A., & Nahrstedt, K. (Eds.). (1997). *Building QoS into distributed systems*. Boston: Kluwer.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

CASIDE. (2005). *CASIDE Project (EP/C005589): Investigating cooperative applications in situated display environments*. Retrieved March 13, 2006, from http://www.caside.lancs.ac.uk/

CERN (2001). *European organisation for nuclear research*. Retrieved March 13, 2006, from http://public.web.cern.ch/Public/

Chalmers, M., Bell, M., Brown, B., Hall, M., Sherwood, S., & Tennent, P. (2005). Gaming on the edge: Using seams in Ubicomp games. In *Proceedings of ACM Advances in Computer Entertainment* (*ACE05*) (pp. 306–309). New York: ACM Press.

Clarke, D., Dix, A., Ramduny, D., & Trepess, D. (Eds.). (1997, June). *Workshop on time and the web*. Staffordshire University. Retrieved March 13, 2006, from http://www.hiraeth.com/conf/web97/papers/

Dix, A. J. (1987). The myth of the infinitely fast machine. In *People and Computers III—Proceedings of HCI'87* (pp. 215–228). Cambridge, UK: Cambridge University Press.

Dix, A. J. (1991). *Formal methods for interactive systems*. New York: Academic Press.

Dix, A. J. (1994). *Seven years on, the myth continues*. Research Report-RR9405, Huddersfield, U.K.: University of Huddersfield. Retrieved March 13, 2006, from http://www.hcibook.com/alan/papers/myth95/7-years-on.html

Dix, A. J. (1995). Dynamic pointers and threads. *Collaborative Computing, 1*(3), 191–216.

Dix, A. J. (1996). Closing the Loop: modelling action, perception and information. In T. Catarci, M. F. Costabile, S. Levialdi, & G. Santucci (Eds.), *Proceedings of AVI'96—Advanced Visual Interfaces, Gubbio, Italy* (pp. 20–28). New York: ACM Press.

Dix, A. (2001a). artefact + marketing = product. *Interfaces, 48*, 20–21. London: BCS-HCI Group. Retrieved March 13, 2006, from http://www.hiraeth.com/alan/ebulletin/product-and-market/

Dix, A. (2001b). *Cyber-economies and the real world*. Keynote—South African Institute of Computer Scientists and Information Technologists Annual Conference, SAICSIT 2001, Pretoria, South Africa. Retrieved Septmeber 25–20, 2001, from http://www.hcibook.com/alan/papers/SAICSIT2001/

Dix, A., & Brewster, S. A. (1994). Causing trouble with buttons. In D. England (Ed.), *Ancilliary Proceedings of HCI'94, Glasgow, Scotland*. London: BCS-HCI Group. Retrieved March 13, 2006, from http://www.hcibook.com/alan/papers/buttons94/

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). *Human–computer interaction* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall, Inc.

Dix, A., Rodden, T., Davies, N., Trevor, J., Friday, A., & Palfreyman, K. (2000). Exploiting space and location as a design framework for interactive mobile systems. *ACM Transactions on Computer–Human Interaction* (*TOCHI*), *7*(3), 285–321.

Dodge, M., & Kitchin, R. (2001). *Atlas of Cyberspace*. Reading, MA: Addison Wesley. http://www.cybergeography.org/atlas/

Dourish, P., & Bellotti, V. (1992). Awareness and Coordination in Shared Workspaces. In *Proceedings of CSCW'92* (pp. 107–114). New York: ACM Press

Edwards, W. K., & Rodden, T. (2001). *Jini, example by example*. Upper Saddle River, NJ: SUN Microsystems Press.

Electrolux. (1999). *Screenfridge.* Retrieved March 13, 2006, from http://www.electrolux.com/screenfridge/

Ellis, C. A., & Gibbs, S. J. (1989). Concurrency control in groupware systems. In *Proceedings of 1989 ACM SIGMOD International Conference on Management of Data, SIGMOD Record, 18*(2), 399–407.

FCC. (2006). *Voice over Internet Protocol*. Federal Communications Commission. Retrieved March 13, 2006, from http://www.fcc.gov/voip/

Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., et al. (2001). What makes websites credible? A report on a large quantitative study. *Proceedings of CHI2001,* Seattle, 2001, Also *CHI Letters, 3*(1), 61–68. New York: ACM Press.

Foster, I. (2000). Internet computing and the emerging grid. *Nature*, *webmatters*. Retrieved March 13, 2006, http://www.nature.com/nature/webmatters/grid/grid.html

Foster, I., & Kesselman, C. (Eds.). (1999). *The grid: Blueprint for a new computing infrastructure.* San Francisco: Morgan-Kaufmann.

Gaver, W. W., Smith, R. B., & O'Shea, T. (1991). Effective sounds in complex situations: The ARKola simulation. In S. P. Robertson, G. M. Olson, & J. S. Olson (Eds.), *Reaching through technology—CHI'91 conference proceedings* (pp. 85–90). New York: ACM Press.

Gellersen, H-W., Beigl, M., & Krull, H. (1999). The MediaCup: Awareness technology embedded in an everyday object. In H.-W. Gellersen (Ed.), *Handbeld & ubiquitous computing, Lecture notes in computer science* (Vol. 1707; pp. 308–310). Berlin, Germany: Springer.

Gray, W. D., John, B. E., & Atwood, M. E. (1992). The precis of project ernestine or an overview of a validation of goms. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), *Striking a balance, Proceedings of the CHI'92 Conference on human factors in computing systems* (pp. 307–312). New York: ACM Press.

Greenhalgh, C. (1997). Analysing movement and world transitions in virtual reality tele-conferencing. In J. A. Hughes, W. Prinz, T. Rodden, & K. Schmidt (Eds.), *Proceedings of ECSCW 97* (pp. 313–328). Dordrecht, The Netherlands: Kluwer Academic Publishers.

GRID. (2001). *GRID Forum home page*. Retrieved March 13, 2006, from http://www.gridforum.org/

Gutwin, C., Benford, S., Dyck, J., Fraser, M., Vaghi, I., & Greenhalgh, C. (2004). Revealing delay in collaborative environments. In *Proceedings of ACM Conference on Computer–Human Interaction, CHI'04* (pp. 503–510). New York: ACM Press.

Halasz, F., Moran, T., & Trigg, R. (1987). NoteCards in a nutshell. In *Proceedings of the CHI + GI* (pp. 45–52). New York: ACM Press.

HFIG (2001). *Hybrid feature interaction group home page*, Retrieved March 13, 2006, from University of Glasgow, Scotland website http://www.dcs.gla.ac.uk/research/hfig/

Hindus, D., Mainwaring, S. D., Leduc, N., Hagström, A. E., & Bayley, O. (2001). Casablanca: designing social communication devices for the home. In *Proceedings of CHI 2001* (pp. 325–332). New York: ACM Press.

Howard, S., & Fabre, J. (Eds.) (1998). Temporal aspects of usability [Special issue]. *Interacting with Computers, 11*(1), 1–105.

Hughes, D., Gibson, S., Walkerdine, J., & Coulson, G. (2006, February). Is deviant behaviour the norm on P2P file-sharing networks? In *IEEE Distributed Systems Online*, *7*(2). Retrieved March 13, 2006, from http://dsonline.computer.org

ICQ (2001). *ICQ home page*. Retrieved March 13, 2006, from http://www.icq.com/products/whatisicq.html

IEEE (2001). *IEEE 802.11 working group*. Retrieved March 13, 2006, from http://www.ieee802.org/11/

IPng (2001). IP Next generation working group home page. Retrieved March 3, 2007, title from web page is: *IP Version 6* (IPv6). Retrieved March 13, 2006, from http://playground.sun.com/pub/ipng/html

ISO/IEC 7498. (1994). Information technology—Open systems interconnection—Basic reference model: The basic model. International standards organisation. Retrieved March 3, 2007, http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In J. C. Chew & J. Whiteside (Eds.), *Empowering people—Proceedings of CHI'90 human factors in computer systems* (pp. 107–115). New York: ACM Press.

Johnson, C. (1997). What's the web worth? The impact of retrieval delays on the value of distributed information. In D. Clarke, A. Dix, D. Ramduny, & D. Trepress (Eds), *Workshop on Time ad the web,* Retrieved March 13, 2006, from http://www.hiraeth.com/conf/web97/papers/johnson.html

Johnson, C., & Gray, P. (Eds.) (1996). Temporal Aspects of Usability, report of workshop in Glasgow, June 1995, *SIGCHI Bulletin, 28*(2), 32–61. New York: ACM Press. http://www.acm.org/sigchi/bulletin/1996.2/timeintro.html

JPEG (2001). *Joint Photographic Experts Group home page* [On-line]. http://www.jpeg.org/public/jpeghomepage.htm

Kistler, J. J., & Satyanarayanan, M. (1992). Disconnected operation in the CODA file system. *ACM Transactions on Computer Systems, 10*(1), 3–25.

Lavery, D., Kilgourz, A., & Sykeso, P. (1994). Collaborative use of X-Windows applications in observational astronomy. In G. Cockton, S. Draper, & G. Wier (Eds.), *People and computers* (Vol 9; pp. 383–396). Cambridge, UK: Cambridge University Press.

Light, A., & Wakeman, I. (2001). Beyond the interface: Users' perceptions of interaction and audience on websites. In D. Clarke & A. Dix (Eds.), Interfaces for the active web (Part 1) [Special issue]. *Interacting with Computers, 13*(3), 401–426.

Lock, S., Allanson, J., & Phillips, P. (2000). User-driven design of a tangible awareness landscape. In D. Boyahrski & W. Kellogg, *Proceedings of Symposium on Designing Interactive Systems* (pp. 434–440). New York: ACM Press.

Lock, S., Rayson, P., & Allanson, J. (2003). Personality Engineering for Emotional Interactive Avatars. In *Human Computer Interaction, Theory and practice (Part II), Volume 2 of the Proceedings of the 10th International Conference on Human–Computer Interaction* (pp. 503–507). Mahwah, NJ: Lawrence Erlbaum Associates.

Mariani, J. A., & Rodden, T. (1991). The impact of CSCW on database technology. In *Proceedings of ACM Conference on Computer Supported Cooperative Work (includes critique of 'transparency' in a CSCW setting)*. New York: ACM Press.

Mauve, M. (2000). Consistency in replicated continuous interactive media. In *Proceedings of CSCW'2000* (pp. 181–190). New York: ACM Press.

McDaniel, S. E., & Brinck, T. (1997). Awareness in Collaborative Systems: A CHI 97 Workshop (report). In *SIGCHI Bulletin, 29*(4). New York: ACM Press. Workshop report: http://www.acm.org/sigchi/bulletin/1997.4/mcdaniel.html Workshop web pages: http://www.usabilityfirst.com/groupware/awareness/workshop/

McManus, B. (1997). *Compensatory actions for time delays*. In Clarke, http://www.hiraeth.com/conf/web97/papers/barbara.html

Millett, L. I., Friedman, B., & Felten, E. (2001). Cookies and web browser design: toward informed consent online. In *Proceedings of CHI2001, CHI Letters, 3*(1), 46–52.

Mitsopoulos, E. (2000). A Principled Approach to the Design of Auditory Interaction in the Non-Visual User Interface. *DPhil Thesis*, University of York, UK. http://www.cs.york.ac.uk/ftpdir/reports/YCST-2000 07.zip

MPEG. (2001). *Moving Picture Experts Group home page* [On-line]. http://www.cselt.it/mpeg/

Olson, M., & Bly, S. (1991). The Portland experience: A report on a distributed research group. *International Journal of Man-Machine Studies, 34*(2), 11–228.

Palfreyman, K., & Rodden, T. (1996). A protocol for user awareness on the World Wide web. In *Proceedings of CSCW'96* (pp. 130–139). New York: ACM Press.

Patterson, J. F., Day, M., & Kucan, J. (1996). Notification servers for synchronous groupware. In *Proceedings of CSCW'96* (pp. 122–129). New York: ACM Press.

Pausch, R. (1991). Virtual reality on five dollars a day. In S. P. Robertson, G. M. Olson & J. S. Olson (Eds.), *CHI'91 Conference Proceedings* (pp. 265–270). Reading, MA: Addison Wesley.

Ramduny, D., & Dix, A. (1997). Why, What, Where, When: Architectures for Co-operative work on the WWW. In H. Thimbleby, B. O'Connaill, & P. Thomas (Eds.), *Proceedings of HCI'97* (pp. 283–301). Berlin, Germany: Springer.

Ramduny, D., Dix, A., & Rodden, T. (1998). Getting to Know: the design space for notification servers. In *Proceedings of CSCW'98* (pp. 227–235). New York: ACM Press. http://www.hcibook.com/alan/papers/GtK98/

Resnick, P., & Varian, H. R. (Eds.). (1997). *Communications of the ACM* [Special issue on recommender systems], *40*(3), 56–89.

Rodden, T. (1996). Populating the application: a model of awareness for cooperative applications. In M. S. Ackerman (Ed.), *Proceedings of the 1996 ACM Conference on Computer-Supported Cooperative Work* (pp. 87–96). New York: ACM Press.

Rohs, M., Sheridan, J. G., & Ballagas, R. (2004). Direct manipulation techniques for large displays using camera phones. In *Proceedings of 2nd International Symposium on Ubiquitous Computing Systems*. Retrieved March 13, 2006, from http://www.equator.ac.uk/index.php/articles/113

Roussel, N. (1999). Beyond webcams and videoconferencing: Informal video communication on the web. In *Proceedings of The Active web*. Retrieved March 13, 2006, from http://www.hiraeth.com/conf/activeweb/

Roussel, N. (2001). Exploring new uses of video with videoSpace. In *Proceedings of EHCI'01, the 8th IFIP Working Conference on Engineering for Human-Computer Interaction, Lecture Notes in Computer Science* (pp.73–90). Berlin: Springer-Verlag. Retrieved March 13, 2006), from http://www-iiuf.unifr.ch/%7erousseln/publications/EHCI01.pdf

Sandor, O., Bogdan, C., & Bowers, J. (1997). Aether: An Awareness Engine for CSCW. In J. Hughes (Ed.), *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work* (ECSCW'97) (pp. 221–236). Dordrecht, The Netherlands: Kluwer Academic.

Sangha, A. (2001, June). *Legal implications of location based advertising*. Interview for the WAP Group. Retrieved March 13, 2006, from http://www.thewapgroup.com/53762_1.DOC

Sarvas, R., Herrarte, E., Wilhelm, A., & Davis, M. (2004). Metadata creation system for mobile images. In *Proceedings of Second International Conference on Mobile Systems, Applications, and Services* (pp. 36–48). New York: ACM Press.

Semacode. (2006). *Semacode*. Retrieved March 13, 2006, from http://semacode.org/

Schneier, B. (1996). *Applied cryptography* (2nd ed.). New York: Wiley.
SETI@home. (2001). *SETI@home—the search for extra-terrestrial intelligence*. http://setiathome.ssl.berkeley.edu/

Shneiderman, B. (1984). Response time and display rate in human performance with computers. *ACM computing surveys, 16*(3), 265–286. Retrieved March 13, 2006, New York: ACM Press.

Siegal, D. (1999). *Futurize your enterprise*. New York: Wiley. http://www.futurizenow.com

Skype. (2006). *Skype* [On-line]. http://www.skype.com/

Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revisited: early experiences with multiuser interfaces. *ACM Transactions on Office Information Systems, 5*(2), 147–167.

Stevens, W. R. (1998). *UNIX network programming, Networking APIs: Sockets and XTI* (Vol. 1). Englewood Cliffs, NJ: Prentice Hall.

Stevens, W. R. (1999). *UNIX network programming, Interprocess communications* (2nd ed., Vol. 2). NJ: Prentice Hall.

Stifelman, L., Arons, B., & Schmandt, C. (2001). The audio notebook: Paper and pen interaction with structured speech. In *Proceedings of CHI2001, CHI Letters, 3*(1), 182–189.

Sun, C., & Ellis, C. (1998). Operational transformation in real-time group editors: Issues, algorithms, and achievements. In *Proceedings of CSCW'98* (pp. 59–68). New York: ACM Press.

SUN Microsystems. (2001). *Awarenex*. Retrieved March 13, 2006, from http://www.sun.com/research/features/awarenex/

Taylor, A., Izadi, S., Swan, L., Harper, R., & Buxton, W. (2006). *Building Bowls for Miscellaneous Media. Physicality 2006 workshop*, Lancaster University, 6–7 Feb. 2006 [On-line]. http://www.physicality.org/

thePooch (2006) .:*thePooch*:. Retrieved March 13, 2006, from http://www.thepooch.com/

Tillotson, J. (2005). *Scent whisper*. Central Saint Martins College of Art & Design. Exhibited in SIGGRAPH CyberFashion 0100. Retrieved March 13, 2006), from http://psymbiote.org/cyfash/2005/

Toye, E., Madhavapeddy, A., Sharp, R., Scott, D., Blackwell, A., & Upton, E. (2004). *Using camera-phones to interact with context-aware mobile services* (Technical Report UCAM-CL-TR-609). Retrieved (date) from University of Cambridge, Computer Laboratory website. Retrieved March 13, 2006, from http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-609.pdf

Vidot, N., Cart, M., Ferriz, J., & Suleiman, M. (2000). Copies convergence in a distributed real-time collaborative environment. In *Proceedings of CSCW'2000* (pp. 171–180). New York: ACM Press.

Wikipedia. (2006). File sharing. *Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=File_sharing&oldid=34285993.

WiMedia. (2006). *WiMedia alliance*. Retrieved March 13, 2006, from http://wimedia.org/

Wisneski, G., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., et al. (1998). Ambient Display: Turning Architectural Space into an Interface between People and Digital Information. In *Proceedings of the First International Workshop on Cooperative Buildings (CoBuild'98), Lecture Notes in Computer Science* (Vol. 1370; pp. 22–32). Heidelberg: Springer-Verlag.

Witten, I. H., Thimbleby, H. W., Coulouris, G., & Greenberg, S. (1991). Liveware: A new approach to sharing data in social networks. *International Journal of Man–Machine Studies, 34*, 337–348.

Xia, S., Sun, D., Sun, C., Chen, D., & Shen, H. (2004). Leveraging single-user applications for multi-user collaboration: the CoWord approach. In *Proceedings of ACM 2004 Conference on Computer Supported Cooperative Work* (pp. 162–171). Retrieved March 13, 2006, New York: ACM Press.

Zigbee. (2006). *ZigBee Alliance*. http://zigbee.org/

Zimmerman, T. G. (1996). Personal area networks: Near-field intrabody communication. *IBM Systems Journal, 35*(3/4), 609–617. Retrieved March 13, 2006, from http://isj.www.media.mit.edu/projects/isj/SectionE/609.htm

•15•

WEARABLE COMPUTERS

Dan Siewiorek and Asim Smailagic
Carnegie Mellon University

Thad Starner
Georgia Institute of Technology

Introduction **296**
VuMan 3 Maintenance Inspection Wearable Computer **296**
The Wearable Computing CAMP **297**
Corporal: Design Guides for Wearability **297**
Attention **299**
Manipulation **300**
VuMan 3 Dials Pointing 300
Mobile Keyboards 303
Speech Interfaces 304
Vocollect 304
Navigator Wearable Computer with Speech Input . 304
Speech Translation 307
Performance Evaluation **309**
Dual Purpose Speech 310
Perception 310
Conclusion and Future Challenges **311**
References **311**

INTRODUCTION

Computers have become a primary tool for office workers, allowing them to access the information they need to perform their jobs; however, accessing information is more difficult for more mobile users. With current computer interfaces, the user must focus both physically and mentally on the computing device instead of the environs. In a mobile environment, such interfaces may interfere with the user's primary task. Many mobile tasks could, however, benefit from computer support. Our focus is the design of wearable computers that augment, instead of interfere, with the user's tasks. Carnegie Mellon University's VuMan 3 project provides an example of how the introduction of wearable computing to a task can reap many rewards.

VUMAN 3 MAINTENANCE INSPECTION WEARABLE COMPUTER

Many maintenance activities begin with an inspection, during which problems are identified. Job orders and repair instructions are generated from the results of the inspection. The VuMan 3 wearable computer was designed for streamlining Limited Technical Inspections (LTI) of amphibious tractors for the U.S. Marines at Camp Pendleton, California (Smailagic, Siewiorek, Martin, & Stivoric, 1998). The LTI is a 600-element, 50-page checklist that usually takes four to six hours to complete. The inspection includes an item for each part of the vehicle (e.g., front left track, rear axle, windshield wipers, etc.). VuMan 3 created an electronic version of this checklist. The system's interface was arranged as a menu hierarchy and a physical dial and selection buttons controlled navigation. The top level consisted of a menu that gave a choice of function. Once the inspection function was chosen, the component being inspected was selected by its location on the vehicle. At each stage, the user could go up one level of the hierarchy.

The inspector selects one of four possible options about the status of the item: (a) Serviceable, (b) Unserviceable, (c) Missing, or (d) On Equipment Repair Order (ERO). Further explanatory comments about the item can be selected (e.g., the part is unserviceable due to four missing bolts).

The LTI checklist consists of a number of sections, with approximately one hundred items in each section. The user sequences through each item by using the dial to select "Next Item," or "Next Field." A "smart cursor" helps automate some of the navigation by positioning the user at the next most likely action.

As part of the design process, a field study was performed. In typical troubleshooting tasks, one Marine would read the maintenance manual to a second Marine, who performs the inspection. With the VuMan 3, only one Marine is needed for the task, as that Marine has the electronic maintenance manual. Thus, the physical manual does not have to be carried into hard-to-reach places.

The most unanticipated result was a 40% reduction in inspection time. The bottom right image of Fig. 15.1 demonstrates

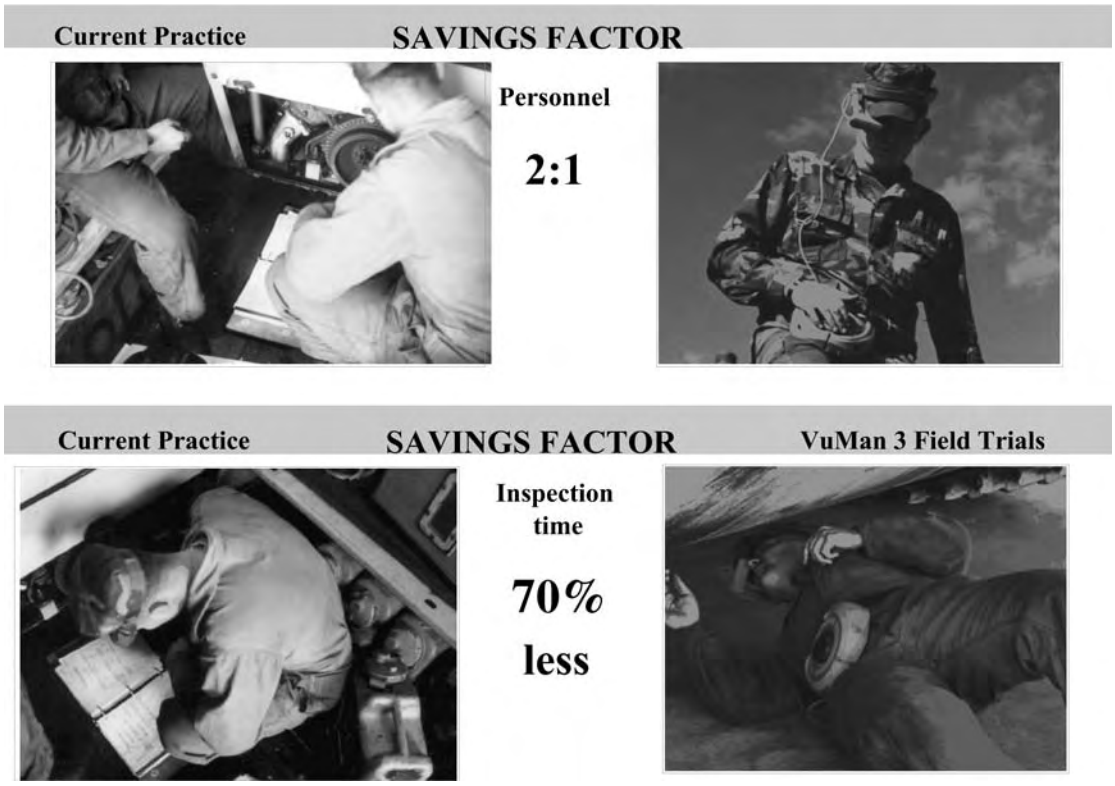


FIGURE 15.1. VuMan 3 Savings Factors.

the reason for this result. Here, the Marine is on his or her side looking up at the bottom of the amphibious tractor. In such places, it is hard to read or write on the clipboard typically used for inspections. The Marine constantly gets into position, crawls out to read instructions, crawls back into position for the inspection, and then crawls out again to record the results. In addition, the Marine tends to do one task at a time when the Marine might have five things to inspect in one place. This extra motion has a major impact on the time required to do a task. By making information truly portable, wearable computers can improve the efficiency of this application and many other similar ones.

The second form of time savings with the VuMan 3 occurred when the inspection is finished. The wearable computer requires a couple of minutes to upload its data to the logistics computer. The manual process, however, required a typist to enter the Marine's handwritten text into the computer. Given that the soldier may have written the notes in cold weather while wearing gloves, the writing may require some interpretation. This manual process represents another 30% of the time.

Such redundant data entry is common when users are mobile (Starner, Snoeck, Wong, & McGuire, 2004). There are numerous checklist-based applications including plant operations, preflight checkout of aircraft, inventory, and so forth that may benefit from a form-filling application run on a wearable computer. In the case of the VuMan 3 project, the results were striking. From the time the inspection was started until the data was entered into the logistics computer, 70% of the time was saved by using the wearable. There was a potential savings by reducing maintenance crews from two to one. Finally, there was also a savings in weight over paper manuals.

THE WEARABLE COMPUTING CAMP

Designing wearable computer interfaces requires attention to many different factors due to their closeness to the body and their use while performing other tasks. For the purposes of discussion, we have created the "CAMP" framework, which consists of the following factors:

- **Corporal**: Wearables should be designed to interface physically with the user without discomfort or distraction.
- **Attention**: Interfaces should be designed for the user's divided attention between the physical and virtual worlds.
- **Manipulation**: When mobile, users lose some of the dexterity assumed by desktop interfaces. Controls should be quick to find and simple to manipulate.
- **Perception**: A user's ability to perceive displays, both visual and audio, is also reduced while mobile. Displays should be simple, distinct, and quick to navigate.

Power, heat, on-body, and off-body networking, privacy, and many other factors also affect on-body computing (Starner, 2001). Many of these topics are the subjects of current research, and much work will be required to examine how these factors interrelate. Due to space, we will concentrate mainly on CAMP principles and practice in the remainder of this chapter.

CORPORAL: DESIGN GUIDES FOR WEARABILITY

The term *wearable* implies the use of the human body as a support environment for the object described. Society has historically evolved its tools and products into more portable, mobile, and wearable form factors. Clocks, radios, and telephones are examples of this trend. Computers are undergoing a similar evolution. Simply shrinking computing tools from the desktop paradigm to a more portable scale does not take advantage of a whole new context of use. While it is possible to miniaturize keyboards, human evolution has not kept pace by shrinking our fingers. There is no Moore's Law for humans. The human anatomy introduces minimal and maximal dimensions that define the shape of wearable objects, and the mobile context also defines dynamic interactions. Conventional methods of interaction, including keyboard, mouse, joystick, and monitor, have mostly assumed a fixed physical relationship between user and device. With wearable computers, the user's physical context may be constantly changing. Symbol's development of a wearable computer for shipping hubs provides an example of how computing must be adapted for the human body.

As a company, Symbol is well known for its barcode technology; however, it is also one of the first successful wearable computer companies, having sold over 100,000 units from its WSS 1000 line of wearable computers (see Fig. 15.2). The WS-1000 consists of a wrist-mounted wearable computer that features a laser barcode scanner encapsulated in a ring worn on the user's finger. This configuration allows the user to scan barcodes while keeping both hands free to manipulate the item being scanned. Because the user no longer has to fumble with a desk-tethered scanner, these devices increase the speed at which the user can manipulate packages and decrease the overall strain on the user's body. Such features are important in shipping hubs, where millions of packages are scanned by hand every year. Symbol spent over U.S. $5 million and devoted 40,000 hours of



FIGURE 15.2. Symbol's WSS 1000 series wrist-mounted wearable computer with ring scanner.

testing to develop this new class of device, and one of the major challenges was adapting the computer technology to the needs of the human body (Stein, Ferrero, Hetfield, Quinn, & Krichever, 1998).

One of the first observations made was that users may be widely varying shapes and sizes. Specifically, Symbol's scanner had to fit the fingers of both large men and small women. Similarly, the wrist unit had to be mounted on both large and small wrists. Even though the system's wires were designed to be unobtrusive, the system must be designed to break away if entangled and subjected to strain. This policy provided a safeguard for the user.

Initial testing discovered other needs that were obvious in hindsight. For example, the system was strapped to the user's forearm while the user exerted him- or herself moving boxes. Soon, the "soft-good" materials, which were designed for the comfort of the user, became sodden with sweat. After one shift, the user was expected to pass the computer to the operator on the next shift. Not only was the sweat-laden computer mount considered "gross," it also presented a possible health risk. This problem was solved by separating the computer mount from the computer itself. Each user received his or her own mount, which could be kept adjusted to the user's own needs. After each shift, the computer could be removed from the user's mount and placed in the replacement user's mount.

Another unexpected discovery is that the users tended to use the computer as body armor. When a shipping box would begin to fall on the user, the user would block the box with the computer mounted on her or his forearm, as that was the least sensitive part of the user's body. Symbol's designers were surprised to see users adapt their work practices to use the rigid forearm computer to force boxes into position. Accordingly, the computer's case was designed out of high impact materials; however, another surprise came with longer term testing of the computer.

Employees in the test company's shipping hubs constantly reached into wooden crates to remove boxes. As they reached into the crates, the computer would grind along the side. After extended use, holes would appear in the computer's casing, eventually damaging the circuitry. Changing the composition of the casing to be resistant to both abrasion and impact finally fixed the problem.

After several design cycles, Symbol presented the finished system to new employees in a shipping hub. After a couple of weeks' work, test results showed that the new employees felt the system was cumbersome, whereas established employees who had participated in the design of the project felt that the wearable computer provided a considerable improvement over the old system of package scanning. After consideration, Symbol's engineers realized that these new employees had no experiential basis for comparing the new system to the past requirements of the job. As employees in shipping hubs are often short term, a new group of employees were recruited. For two weeks, these employees were taught their job using the old system of package scanning: the employee would reach into a crate, grasp a package, transfer it to a table, grasp a handheld scanner, scan the package, replace the scanner, grasp the package, and transfer it to its appropriate conveyer belt. The employees were then introduced to the forearm-mounted WS-1000. With the wearable computer, the employee would squeeze her or his index and middle finger together to trigger the ring-mounted scanner to scan the package while reaching for it, grasp the package, and transfer it to the appropriate convey belt in one fluid motion. These employees returned very positive scores for the wearable computer.

This lesson, that perceived value and comfort of a wearable computer is relative, was also investigated by Bodine and Gemperle (2003). In short interviews, users were fitted with a backpack or armband wearable and told that the system was either a police monitoring device (similar to those used for house arrest), a medical device for monitoring health, or a device for use during parties. The subjects were then asked to rate the devices on various scales of desirability and comfort. Not surprisingly, the police wearable was considered the least desirable; however, the police function elicited more negative *physical* comfort ratings, and the medical function elicited more positive physical comfort ratings, even though they were the same device. In other words, perceived comfort can be affected by the supposed function of the device.

Researchers have also explored wearability in more general terms. Wearability is defined as the interaction between the human body and the wearable object. Dynamic wearability includes the human body in motion. Design for wearability considers the physical shape of objects and their active relationship with the human form. Gemperle, Kasabach, Stivoric, Bauer, and Martin (1998) explored history and cultures including topics such as clothing, costumes, protective wearables, and carried devices (Siewiorek, 2002). They studied physiology, biomechanics, and the movements of modern dancers and athletes. Drawing upon the experience of CMU's wearables group, over two dozen generations of machines representing over a hundred person years of research, they codified the results into guidelines for designing wearable systems. These results are summarized in Table 15.1.

This team also developed a set of wearable forms to demonstrate how wearable computers might be mounted on the body. Each of the forms was developed by applying design guidelines and follows a simple pattern for ensuring wearability. The pods were designed to house electronic components. All of the forms are between 3/8″ and 1″ thick, and flexible circuits can fit comfortably into the 1/4″ thick flex zones. Beginning with acceptable areas and the humanistic form language, the team considered human movement in each individual area. Each area is unique, and some study of the muscle and bone structure was required along with common movements. Perception of size was studied for each individual area. For testing, minimal amounts of spandex were stretched around the body to attach the forms. The results are shown in Fig. 15.3.

These studies and guidelines provide a starting point for wearable systems designers; however, there is much work to be done in this area. Weight was not considered in these studies, nor were the long-term physiological effects such systems might have on the wearer's body. Similarly, fashion can affect the perception of comfort and desirability of a wearable component. As wearable systems become more common and are used for longer periods of time, it will be important to test these components of wearability.

TABLE 15.1. Design for Wearability Attributes.

Attribute	Comments
Placement	Identify where the computer should be placed on the body. Issues include identifying areas of similar size across a population, areas of low movement/flexibility, and areas large in surface area.
Humanistic Form Language	The form of the object should work with the dynamic human form to ensure a comfortable fit. Principles include inside surface concave to fit body, outside surface convex to deflect objects, tapering sides to stabilize form on body, and radiusing edges/corners to provide soft form.
Human Movement	Many elements make up a single human movement: mechanics of joints, shifting of flesh, and the flexing and extending of muscles and tendons beneath the skin. Allowing for freedom of movement can be accomplished in one of two ways: (a) by designing around the more active areas of the joints, or (b) by creating spaces on the wearable form into which the body can move.
Human Perception of Size	The brain perceives an aura around the body. Forms should stay within the wearer's intimate space, so that perceptually they become a part of the body. The intimate space is between zero and five inches off the body and varies with position on the body.
Size Variations	Wearables must be designed to fit many types of users. Allowing for size variations is achieved in two ways: (a) static anthropometric data, which details point to point distances on different sized bodies, and (b) consideration of human muscle and fat growth in three dimensions using solid rigid areas coupled with flexible areas.
Attachment	Comfortable attachment of forms can be created by wrapping the form around the body, rather than using single point fastening systems such as clips or shoulder straps.
Contents	The system much have sufficient volume to house electronics, batteries, and so forth that, in turn, constrains the outer form.
Weight	The weight of a wearable should not hinder the body's movement or balance. The bulk of the wearable object weight should be close to the center of gravity of the human body, minimizing the weight that spreads to the extremities.
Accessibility	Before purchasing a wearable system, walk and move with the wearable object to test its comfort and accessibility.
Interaction	Passive and active sensory interaction with the wearable should be simple and intuitive.
Thermal	The body needs to breathe and is very sensitive to products that create, focus, or trap heat.
Aesthetics	Culture and context will dictate shapes, materials, textures, and colors that perceptually fit the user and their environment.

ATTENTION

Humans have a finite and nonincreasing capacity that limits the number of concurrent activities they can perform. Herb Simon observed that human effectiveness is reduced as they try to multiplex more activities. Frequent interruptions require a refocusing of attention. After each refocus of attention, a period of time is required to reestablish the context prior to the interruption. In addition, human short-term memory can hold seven plus or minus two (e.g., five to nine) chunks of information. With this limited capacity, today's systems can overwhelm users with data, leading to information overload. The challenge to human computer interaction design is to use advances in technology to preserve human attention and to avoid information saturation.

In the mobile context, the user's attention is divided between the computing task and the activities in the physical environs. Some interfaces, like some augmented realities (Azuma, 1997) and Dual Purpose Speech (Lyons, Skeels, et al., 2004), try to integrate the computing task with the user's behavior in the physical world. The VuMan 3 interface did not tightly couple the virtual and real worlds, but the computer interface was designed specifically for the user's task and allowed the user to switch rapidly between a virtual interface and his or her hands-on vehicle inspection.

Many office productivity tasks such as e-mail or web searching, however, have little relation to the user's environment. The mobile user must continually assess what attentional resources she or he can commit to the interface and for how long before switching attention back to the primary task. Oulasvirta, Tamminen, Roto, and Kuorelahit (2005) specifically examined such situations by fitting cameras to mobile phones and observing users attempting web search tasks while following predescribed routes. Subjects performed these tasks in a laboratory, in a subway car, riding a bus, waiting at a subway station, walking on a quite street, riding an escalator, eating at a cafeteria and conversing, and navigating a busy street. web pages required an average of 16.2 seconds to load and had considerable variance requiring the user to attend the interface. The subjects shifted their attention from the phone interface more often depending on the task: 35% of page loadings in the laboratory versus 80% of the page loadings while walking a quiet street. The duration of continuous attention on the mobile device also varied depending on the physical environment: 8–16 seconds for the laboratory and cafe versus below 6 seconds for the riding the escalator or navigating a busy street. Similarly, the number of attention switches depended on the demands of the environment.

The authors noted that even riding an escalator requires demands on attention (e.g., choosing a correct standing position, monitoring personal space for passers-by, and determining when the end is in order to step off). Accordingly, they are working on a "Resource Competition Framework," based on the Multiple Resource Theory of attention (Wickens, 1984), to relate mobile task demands to the user's cognitive resources. This



FIGURE 15.3. Forms studied for wearability.

framework helps predict when the mobile user will need to adopt attentional strategies to cope with the demands of a mobile task. The authors reported four such strategies observed in their study. The first, "calibrating attention," refers to the process by which the mobile user first attends to the environment and determines the amount of attention the user needs to devote to the environment versus the interface. "Brief sampling over long intervals" refers to the practice of only attending to the environment in occasional, brief bursts to monitor for changes that may require a deviation from plan, such as when reading while walking a empty street. "Task finalization" refers to subjects' preference to finish, when sufficiently close, a task or subtask before switching attention back to the physical environment. "Turntaking capture" occurs when the user is conversing with another person. Attending and responding to another person requires significant concentration, leading to minimal or no attention to the mobile interface.

The third author, who has been using his wearable computer to take notes on his everyday life since 1993, has remarked on similar strategies in his interactions. Describing these attentional strategies and designing interfaces that leverage them will be important in future mobile interfaces. Much research has been performed on aircraft and automobile cockpit design-to-design interfaces that augment, but do not interfere with, the pilot's primary task of navigating the vehicle; however, only recently has it begun to be possible to instrument mobile users and examine interface use (and misuse) "in-the-field" for the mobile computer user (Lyons & Starner, 2001). Now, theories of attention can be applied and tested to everyday life situations.

This newfound ability to monitor mobile workers may help us determine how **not** to design interfaces. In contrast to the VuMan 3 success described above, Ockerman's (2000) PhD thesis "Task Guidance and Procedure Context: Aiding Workers in Appropriate Procedure Following" warns that mobile interfaces, if not properly designed, may hinder the user's primary task. Ockerman studied experienced pilots inspecting their small aircraft before flying. When a wearable computer was introduced as an aid to completing the aircraft's safety inspection checklist, the expert pilots touched the aircraft less (a way many pilots develop an intuition as to the aircraft's condition). In addition, the pilots relied too much on the wearable computer system, which was purposely designed to neglect certain safety steps. The pilots trusted the wearable computer checklist to be complete instead of relying on their own mental checklists. Ockerman showed how such interfaces might be improved by providing more context for each step in the procedure. Another approach would be integrating the aircraft itself into the interface (e.g., use augmented reality to overlay graphics on the aircraft indicating where the pilot physically inspects the plane).

Most recently, DARPA's Augmented Cognition project (Kollmorgen, Schmorrow, Kruse, & Patrey, 2005) aims to create mobile systems that monitor their user's attentional resources and records or delays incoming information in order to present it to the user in a more orderly and digestible time sequence. These systems exploit mobile electroencephalogram (EEG) readings or functional near infrared imaging (fNIR) to monitor the user's brain activations and relate these results to the user's current state. Such projects, if successful on a larger scale, could reveal much about the mental resources required for truly mobile computing.

The Attention Matrix, shown in Fig. 15.4 (Anhalt et al., 2001), categorizes activities by the amount of attention they require. The activities are (a) Information, (b) Communication, and (c) Creation. Individual activities are categorized by the amount of distraction they introduce in units of increasing time: (a) Snap, (b) Pause, (c) Tangent, and (d) Extended. The Snap duration is an activity that is usually completed in a few seconds, such as checking a watch for the time. The user should not have to interrupt their primary activity to perform this activity. The Pause action requires the user to stop their current activity, switch to the new but related activity, and then return to their previous task within a few minutes. Pulling over to the side of the road and checking directions is an example of a pause. A Tangent action is a medium length task that is unrelated to the action in which the user is engaged. Receiving an unrelated phone call is an example of a tangent activity. An Extended action occurs when the user deliberately switches tasks, beginning a wholly new, long-term activity. For the car driver, stopping at a motel and resting for the night is an extended activity.

As distractions on the left of the matrix take less time from the user's primary activity, our intent is to move activities of the matrix towards the left side (Snap). Our goal is to evaluate how this process extends to a larger sample of applications.

MANIPULATION

VuMan 3 Dial Pointing

VuMan 3 added a novel manipulation interface suitable for use when physical attention is occupied. The VuMan3 has a low-resolution display and, consequently, a purely textual interface. Figure 15.5 shows a sample screen from the user interface. The

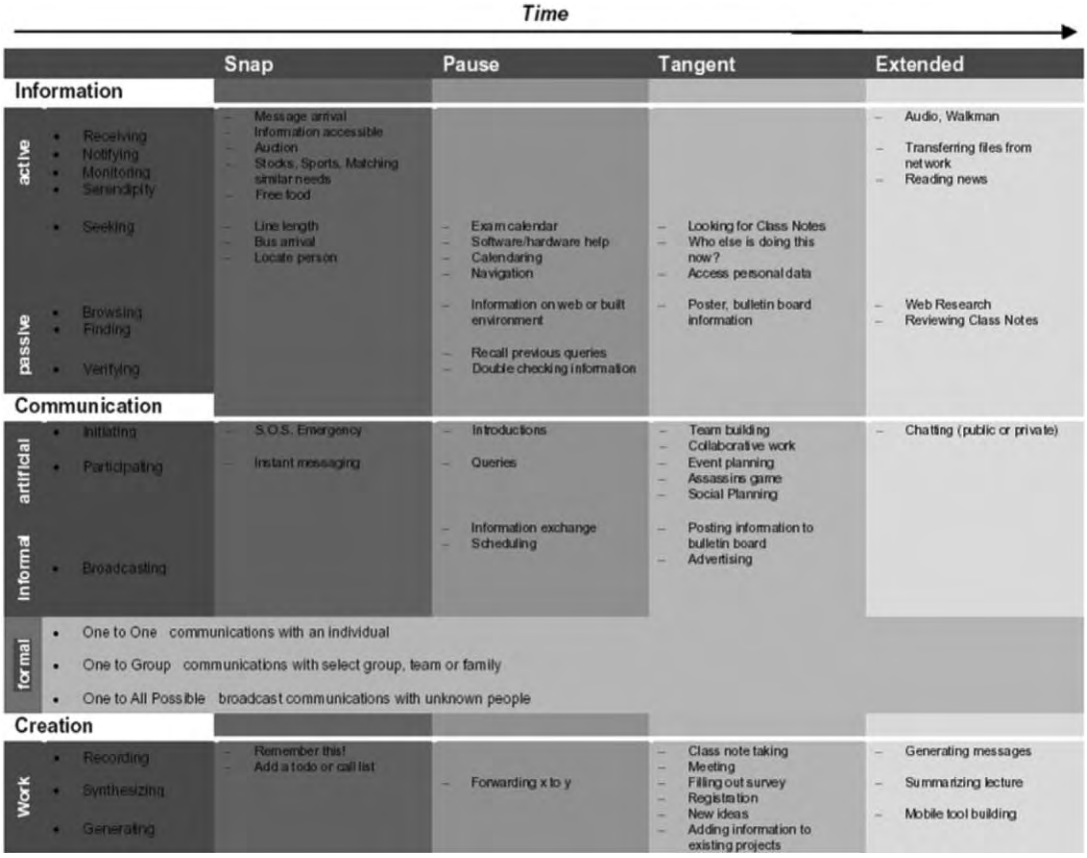


FIGURE 15.4. Attention Matrix.

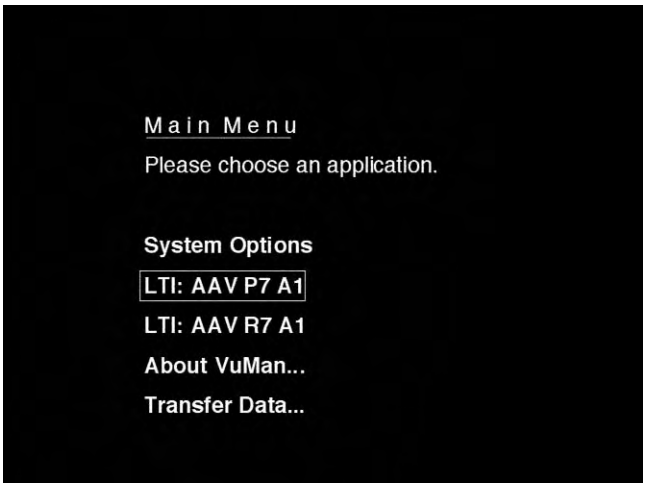


FIGURE 15.5. VuMan 3 Options Screen.

user navigates through a geographically organized hierarchy: top, bottom, front, rear; then left, right, and more detail. Eventually, at the node leafs, individual components are identified. There are over 600 of these components. Each component is indicated to be "Serviceable" or "Unserviceable." If it is serviceable, then no further information is given. If it is unserviceable, then one of a small list of reasons is the next screen.

The user can return up the hierarchy by choosing the category name in the upper right corner, or sequence to the next selection in an ordering of the components. Once a component is marked as serviceable or unserviceable, the next selection in the sequence is automatically displayed for the user. Furthermore, each component has a probability of being serviceable associated with it, and the cursor is positioned over the most likely response for that component.

The screen contains navigational information. Sometimes there is more on a logical screen than can fit on a physical screen. Screen navigation icons are on the left-hand side of the screen. The user can go to the previous physical screen or next physical screen that are functional parts of the logical screen. The user can always go back to the main menu. In Fig. 15.6. the options include the vehicle number, number of hours on the vehicle, vehicle serial number, and so forth, which is used to distinguish this report apart from other reports. The inspection is divided into sections, and different people can be inspecting different sections in parallel. The inspector would pick a section, highlight it by rotating the dial, and then select the highlighted item by pressing a button. The inspector would then receive a detailed set of instructions on how to proceed. In Fig. 15.7, the inspector is instructed to check for damage and bare metal. The "smart cursor" anticipates that the inspector will be filling in

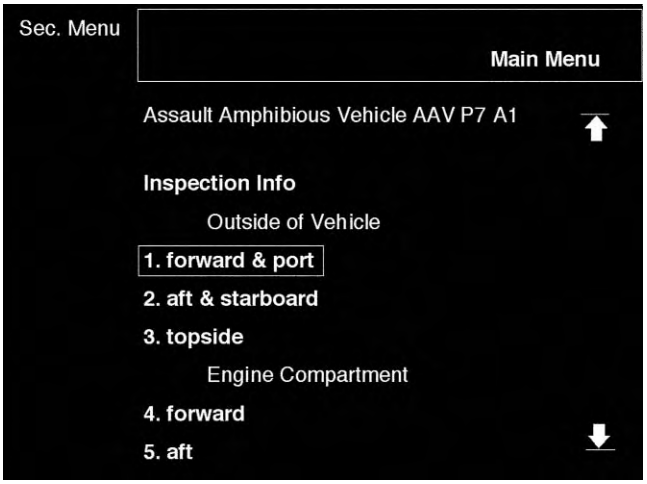


FIGURE 15.6. VuMan 3 Information Screen.



FIGURE 15.7. VuMan 3 Hull Forward Screen.

the "Status Field," whose current value is "none." By clicking, a list of options is displayed, the first of which is "Serviceable." With the Marine LTI, the item is serviceable in 80% of the cases. By ordering the most probable selection first, the interface emulates a paper checklist where most of the items will be checked as "OK." The smart cursor then assumes the most likely step. There is no need to even move the dial—the user merely clicks on the highlighted option. For example, in Fig. 15.8, "Serviceable" has been filled in and the box signifying the next activity is "Next." If all entries are "Serviceable," one would simply tap the button multiple times. If an item is "Unserviceable," the dial is turned and "Unserviceable" is selected. Next, a list of reasons why that particular device was unserviceable would appear. The dial is rotated and one or more of the options are selected. Since more than one reason may be selected as to why it's unserviceable, "Done" is selected to indicate completion. The selected items would appear in the "Comment" field. When the

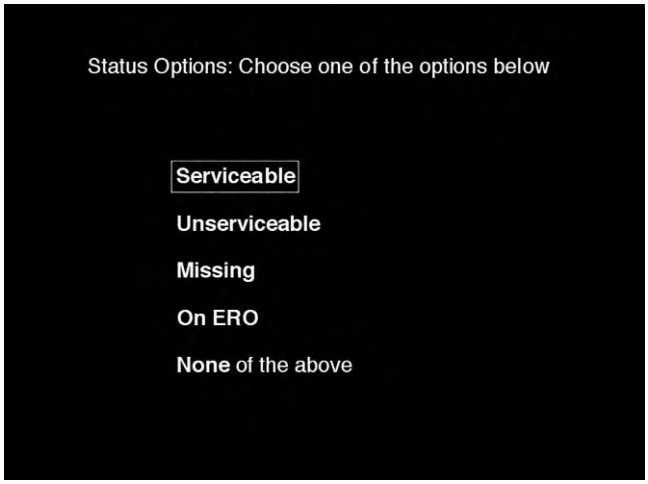


FIGURE 15.8. VuMan 3 Status Screen.

checklist is completed, the data is uploaded to the logistics computer, which would then generate the job work orders.

Several lessons were derived from building the system. As part of the design cycle, a mouse (essentially, a disk with buttons) was tested; however, the physical configuration of the device could be ambiguous. Was the left button in the proper position when the mouse's tail was towards the user or away from the user? Were the buttons supposed to be at the top? The dial removed this orientation ambiguous.

Another design lesson was to minimize cables. An earlier system had a cable connecting the battery, a cable for the mouse, and a cable for the display. These wires quickly became knotted. To avoid this problem, the VuMan 3 design used internal batteries, and the dial was built into the housing. The only remaining wire connected it to the display.

A third lesson was that wearable computers have a minimum footprint that is comfortable for a user's hand. While the keyboards of palmtop computers are getting smaller, evolution has not correspondingly shrunk our fingers. The thickness of the electronics will become thinner. Eventually, it will be as thick as a sheet of plastic or incorporated into clothing; however, the interface will have a minimal footprint. Furthermore, the interface—no matter where it is located on the user's body—is operated in the same way. This is a major feature of the dial. It can be worn on a user's hip or in the small of the user's back. In airplane manufacturing, where workers navigate small spaces, the hip defines the smallest diameter through which the person can enter. Here a shoulder holster is preferred for the wearable computer. The Marines' oversized coverall pockets were an advantage for the system.

The soldiers could drop the computer into their coveralls and operate it through the cloth of the pocket. In terms of simplicity, as well as orientation independence, the dial integrated with the presentation of information on the screen. Everything on the screen could be considered to be on a circular list. In most cases, less than a dozen items on a screen are selectable. This sparse screen is an advantage on a head mounted display where the user may be reading while are moving. The font must

be large enough to read while the screen is bouncing. The dial should be an intuitive interface for web browsing. Probably there are less than a half dozen items on a typical page to select, and it is rotated clockwise or counter clockwise. A button is then used to select the highlighted item. VuMan 3 had three types of buttons that all performed the same function. The buttons support left-hand and right-hand thumb-dominant as well as a central button for finger-dominant users.

Mobile Keyboards

The VuMan 3 addressed the problem of menu selection in the mobile domain and effectively used a 1D dial to create a pointing device that can be used in many different mobile domains. For tasks such as wireless messaging, however, more free-form text entry is needed. While speech technology has made great strides in the last decade, speech recognition is very difficult in the mobile environment and often suffers from high error rates. In addition, speech is often not socially acceptable (e.g., in hospitals, meetings, classrooms, etc.). Keyboard interfaces still provide one of the most accurate and reliable methods of text entry.

Since 2000, wireless messaging has been creating billions of dollars of revenue for mobile phone service providers, and over 1 trillion messages are currently being typed per year. Until recently, many of these messages were created using the Multitap or T9 input method on phone keypads. Yet studies have shown that users average a slow 10–20 words per minute (wpm) using these common typing methods (for comparison, a highly skilled secretary on a desktop averages 70–90 wpm). Given the obvious desire for mobile text input, HCI researchers have begun re-examining keyboards. While keyboard entry has been well studied in the past, mobility suggests intriguing possibilities. For example, if an adequate method of typing can be combined with a sufficient display for the mobile market, computing may move "off-the-desktop" permanently.

Traditionally, text entry studies emphasize learnability, speed, and accuracy; however, a mobile user may not be able to devote all of his or her attention to the text entry process. For example, the user may be taking notes on a conversation and wish to maintain eye contact with his or her conversational partner, or the user may be in a meeting and may hide the keyboard under the desk to avoid distracting others with the keyboard's noise and his or her finger motions. The user might also attempt to enter text while walking and need to attend to the physical environment instead of looking at the screen. These conditions all describe "blind" typing, where the user enters text with only occasional glances at the screen to ensure that the text has been entered correctly.

Lyons, Starner, and colleagues (2004) and Clawson, Lyons, Starner, and Clarkson (2005) have performed longitudinal studies on two keyboards—(a) Handykey's Twiddler (Fig. 15.9) and (b) the mini-QWERTY "thumb" keyboard (Fig. 15.10)—to determine if they might achieve desktop-level text entry in the mobile domain. As the "average" desktop entry rate was considered to be 30 wpm, including hunt-and-peck typists, this benchmark was chosen as the minimum for speed. Traditionally, very high accuracy is desired for desktop typing. As a culture of informal e-mail and SMS messaging has developed, however, less accu-



FIGURE 15.9. Handykey's Twiddler.



FIGURE 15.10. Mini-QWERTY Thumb Keyboard.

rate typing has become common. The community is debating how to reconcile speed and accuracy measures; however, error rates of approximately 5% per character are common in current mobile keyboard studies.

With the Twiddler, novices averaged 4 wpm during the first 20-minute session and averaged 47 wpm after 25 hours of practice (75 20-minute sessions). The fastest user averaged 67 wpm, which is approximately the speed of one of the authors who has been using the Twiddler for twelve years. While 25 hours of practice seems extreme, a normal high school typing class involves almost three times that training time to achieve a goal of 40 wpm.

Even so, mobile computer users may already have experience with desktop QWERTY keyboards. Due to their familiarity with the key layout, these users might more readily adopt a mini-QWERTY keyboard for mobile use. Can a mini-QWERTY keyboard achieve desktop rates? The study performed by Clawson and colleagues (2005) examined the speed and accuracy of experienced desktop typists on two different mini-QWERTY designs. These subjects averaged 30 wpm during the first twenty-minute session and increased to 60 wpm by the end of four hundred minutes of practice!

While both of these studies easily achieved desktop typing rates and had error rates comparable to past studies, can these keyboards be used while mobile? While neither study tested keyboard use while the user was walking or riding in a car, both experimented with blind text entry (in that, in at least one condition, typists could not look at the keyboard nor the output of their typing). When there was a statistically significant difference between blind and normal typing conditions, experienced Twiddler typist slightly improved their speeds and decreased their error rates. Experienced mini-QWERTY typists, however, were significantly inhibited by the blind condition, with speeds of 46 wpm and approximately three times the error rate even after 100 minutes of practice. These results might be expected in that Twiddler users are trained to type without visual feedback from the keyboard, whereas the mini-QWERTY keyboard design assumes that the user can see the keyboard to help disambiguate the horizontal rows of small keys.

The results of these studies demonstrate that there are multiple ways that desktop typing rates can be achieved on a mobile device. The question remains, however, whether the benefits of typing quickly while "blind" or moving will be sufficient to cause users to learn a new text entry method. Other benefits might also affect the adoption of keyboards in the future. For example, a 12-button device such as the Twiddler can be the size of a small mobile phone and still perform well, while 40-button mini-QWERTY keyboards may have already shrunk as much as is possible for users' hands. Another factor may be adoption of mobile computing in developing countries. According to Techweb, almost 1 billon mobile phones were shipped in 2005. Many new mobile phone users will not have learned to type on a Roman alphabet keyboard and may be more concerned with quick learning than compatibility with desktop-input skills.

Speech Interfaces

Vocollect. Mobile keyboards are not suitable for applications in which hands-free control is necessary, such as warehouse applications. Pittsburgh-based Vocollect focuses on package manipulation—in particular, the warehouse-picking problem. In this scenario, a customer places an order consisting of several different items stored in a supplier's warehouse. The order transmits from the warehouse's computer to an employee's wearable computer. In turn, each item and its location are spoken to the employee through a pair of headphones. The employee can control how this list is announced through feedback via speech recognition, and can also report inventory errors as they occur. The employee accumulates the customer's order from the warehouse's shelves and ships it. This audio-only interface also frees the employee to manipulate packages with both hands, whereas a pen-based system would be considerably more awkward. As of December 2000, Vocollect had approximately 15,000 users and revenues between U.S. $10 and $25 million.



FIGURE 15.11. Vocollect's audio-based wearable computer.

Navigator wearable computer with speech input. Boeing has been pioneering "augmented reality" using a head-mounted, see-through display. As the user looks at the aircraft, the next manufacturing step is superimposed on the appropriate portion of the aircraft. One of their first applications is fabrication of wire harnesses. Every aircraft is essentially unique. They may be from different airlines. Even if they are from the same airline, one might be configured for a long-haul route and another for a short-haul route. The airline may specify different configurations. For example, their galleys will be in different places, the wire harnesses would change, and so forth. Wire harnesses are fabricated months before they are assembled into the aircraft. The assembly worker starts with a Peg-board measuring about three feet high and six feet long. Mounted on the board is a full-sized diagram of the final wire harness. Pegs provide support for the bundles of wire as they form. The worker (a) selects a precut wire, (b) reads its identification number, (c) looks up the wire number on a paper list to find the start-

ing coordinates of the wire, (d) searches for the wire on the diagram, and (e) threads the wire following the route on the diagram. With augmented reality, the worker selects a wire and reads the wire identification from the bar code. A head tracker provides the computer with information on where the worker is looking and superimposes the route for that particular wire on the board. Trial evaluations indicate a savings of 25% of the assembly effort primarily due to elimination of cross-referencing the wire with paper lists.

The Navigator 2, circa 1995, was designed for a voice-controlled aircraft inspection application (Siewiorek, Smailagic, & Lee, 1994). The speech recognition system, with a secondary, manually controlled cursor, offers complete control over the application in a hands-free manner, allowing the operator to perform an inspection with minimal interference from the wearable system. Entire—or portions of—aircraft manuals can be brought on-site as needed, using wireless communication. The results of inspection can then be downloaded to a maintenance logistic computer.

Consider one portion of Navigator 2's application, three-dimensional inspections. The application was developed for McClellan Air Force Base in Sacramento, California and the KC-135 aerial refueling tankers. Every five years, these aircraft are stripped down to bare metal. The inspectors use magnifying glasses and pocket knives to hunt for corrosion and cracks.

At startup, Fig. 15.12 the application prompts the user for either their choice of activating the speech recognition system or not. The user then proceeds to the Main Menu. From this location, several options are available, including online documentation, assistance, and the inspection task (Fig. 15.13). Once the user chooses to begin an inspection, information about the inspection is entered, an aircraft type to examine is selected, and the field of interest is narrowed from major features (Left Wing, Right Tail, etc.; Fig. 15.14) to more specific details (individual panes in the cockpit window glass; Fig. 15.15). A coordinate system is superimposed on the inspection region. The horizontal coordinates begin from the nose and the vertical coordinates are "water lines" derived as if the airplane was floating. The inspector records each imperfection in the skin at the corresponding location on the display. The area covered by each defect is recorded, as well as the type of defect, such as Corroded, Cracked, or Missing. To maximize usability, each item or control may be selected simply by speaking its name. Figure 15.16 shows the Navigator 2 systems in use.

The user navigates to the display corresponding to the portion of the skin currently being inspected. This navigation is partially textual based on buttons (choose aircraft type to be inspected) and partially graphical based on side perspectives of the aircraft (choose area of aircraft currently being inspected). The navigation can be performed either through a joystick input device or through speech input. The speech input is exactly the text that would be selected. The positioning of the imperfection is done solely through the joystick, since speech is not well suited for the pointing necessary to indicate the position of the imperfection. As the cursor is moved by the joystick, the coordinates and the type of material represented by the cursor is dis-

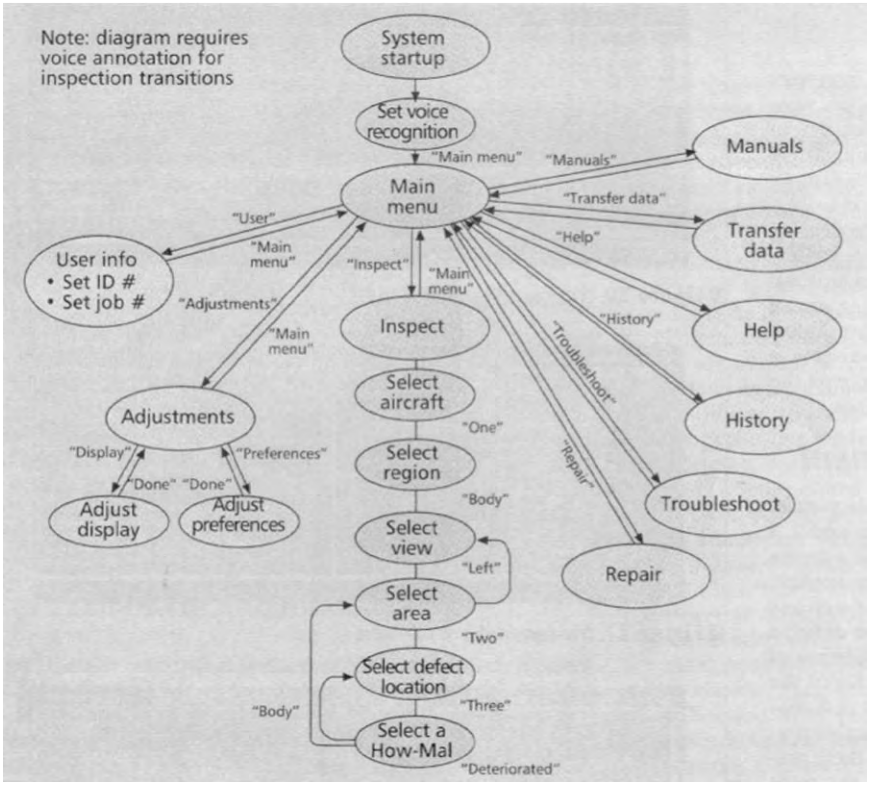


FIGURE 15.12 Navigator 2 Finite State.

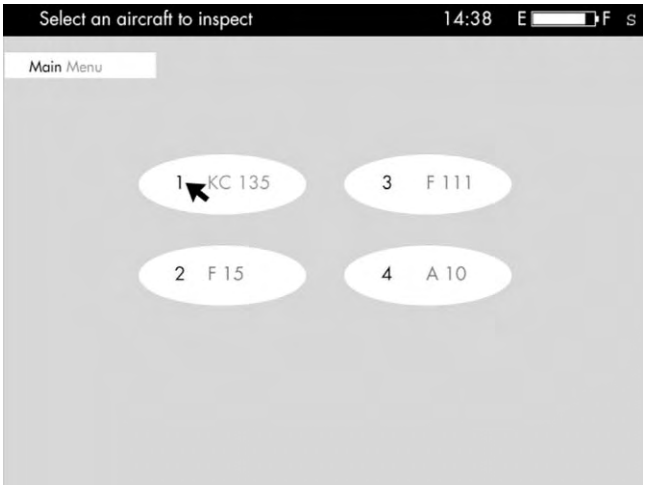


FIGURE 15.13. Navigator 2 Main Menu.

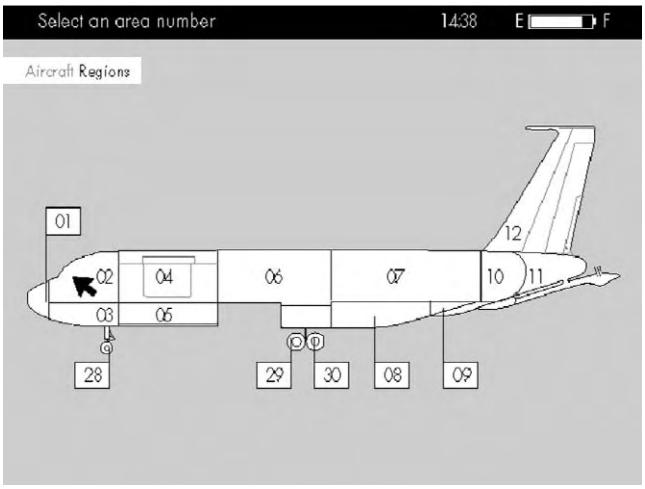


FIGURE 15.14. Navigator 2 Region Selection.

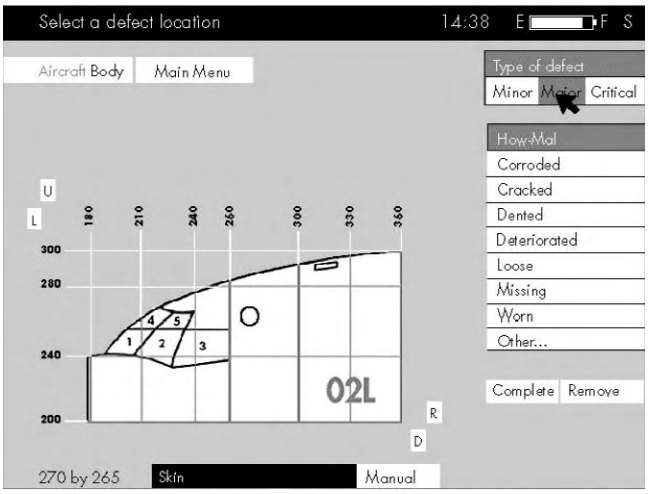


FIGURE 15.15. Sheet metal.



FIGURE 15.16. Navigator 2.

played at the bottom of the screen. If a defect is at the current position, a click produces a list of reasons why that material would be defective, such as corrosion, scratch, and so forth. The defect type can be selected by the joystick or by speaking its name, and the information would go into the database. The user can navigate to the main selection screen by selecting the "Main menu" option on all of the screens. One level up in the hierarchy can also be achieved through a single selection.

The relationship between the user interface design principles and the Navigator 2 user interface is:

- Simplicity of function. The only functions available to the user are to enter skin imperfections for one of four aircrafts, to transfer data to another computer, to enter identification information both for the vehicle and for the inspector, and to see a screen that describes the Navigator 2 project.
- No textual input. The identification information required entering numbers. A special dialogue was developed to enable the entering of numeric information using the joystick as an input device. This was cumbersome for the users, but only needed to be performed once per inspection.
- Controlled navigation. The interface was arranged as a hierarchy. The top level consisted of a menu that gave a choice of function. Once the inspection function and then the vehicle were chosen, the area of the skin inspected was navigated to

via selecting an area of the aircraft to expand. Once an imperfection was indicated, the user had to select one of the allowable types of imperfections. At each stage, the user could go up one level of the hierarchy or return to the main menu.

One of the lessons learned with Navigator 2 is the power of forcing the use of a common vocabulary. Since the average age of the aircraft is 35 years, the types of defects encountered is a very stable set. Previously, one inspector would call a defect "gouged," while another inspector would call the same defect a "scratch." What is the difference between a gouge and a scratch? How much material does it take? How much time does it take to repair? What skill of labor is needed? The logistics problem is much more difficult without a standardized vocabulary. Thus there is a serendipitous advantage in injecting more technology.

A second lesson is that in some cases, the speech recognition front end mistakenly produces the wrong output. Speech recognition systems typically have an error rate of 2% to 10%. The unexpected output may cause the application to produce the wrong result. In one of Navigator 2's early demonstrations, the user was attempting to exit the application; however, the speech recognition system thought a number was spoken. At that point, the application was expecting a second number, but there was no match since the user was saying "Exit," "Quit," "Bye," and so forth. The system appeared to be frozen, when in actuality, there was a mismatch between what the application software was expecting and what the user thought the application state was. The solution was to give the user more feedback on the state of the application by additional on-screen clues. Also developed was a novel application input test generator that took a description of the interface screens and created a list of all possible legal exits from each screen.

A third lesson learned was the criticality of response time. When speech recognition was done in software on Navigator (circa 1995) it was 12 times real time, which became very frustrating. People are less patient when they are on the move than they are when they are at a desktop. People at a desk are willing to wait three minutes for the operating system to boot up, but when a user is on the move, expectations are for instant response like that of portable tools such as a flashlight. For example, some airplanes have a digital computer to control the passengers' overhead lights. It is disconcerting that when the button is pushed, two or three seconds may pass before the light turns on. Even a couple-second delay in a handheld device is disruptive. Users typically continue to push buttons until they receive a response. The extra inputs cause a disconnect between the software and the user. The software receives a stream of inputs, but the user sees outputs that are related to inputs given a long time before the screen actually appears. The situation is similar to listening to yourself talk when there is a second or two delay in the sound played back. Such delays can easily confuse users.

The field evaluation indicated that inspection is composed of three phases. The inspectors would spend the same amount of time maneuvering their cherry picker to access a region of the airplane, visually inspecting and feeling the airplane's skin, and recording the defect's type and location. Navigator 2 reduced the paperwork time by half resulting in an overall time savings of about 18%. Training time to familiarize inspectors with the use of Navigator 2 was about five minutes, after which they would proceed with actual inspections. A major goal of field evaluations is that users perform productive work. They do not want to redo something that was already done once.

The typical inspection, which discovers approximately 100 defects, requires about 36 hours. Today, the inspector takes notes on a clipboard. Upon completion, the inspector fills out forms on a computer. The inspector spends two to three minutes entering each defect. The data entry is thus an additional three-to-four-hour task. Navigator 2 transmits the results of the inspection by radio in less than two minutes.

In summary, evaluations of inspectors before and after the introduction of Navigator 2 indicated a 50% reduction in the time to record inspection information (for an overall reduction of 18% in inspection time) and almost two orders of magnitude reduction in time to enter inspection information into the logistics computer (from over three hours to two minutes). In addition, Navigator 2 weighs two pounds, compared to the cart the inspectors currently use, which carries 25 pounds of manuals.

Speech translation. The SR/LT application (Speech Recognition/Language Translation) consists of three phases: (a) speech-to-text language recognition, (b) text-to-text language translation, and (c) text-to-speech synthesis. The application running on TIA-P (Tactical Information Assistant-Prototype, circa 1996) is the Dragon Multilingual Interview System (MIS), jointly developed by Dragon Systems and the Naval Aerospace and Operational Medical Institute (NAOMI). It is a keyword-triggered, multilingual playback system, which listens to a spoken phrase in English, proceeds through a speech recognition front-end, plays back the recognized phrase in English, and after some delay (~8–10 secs), synthesizes the phrase in a foreign language (Croatian). The other, local person can answer with "Yes," "No," and some pointing gestures. The Dragon MIS has about 45,000 active phrases, in the following domains: medical examination, mine fields, road checkpoints, and interrogation. Therefore, a key characteristic of this application is that it deals with a fixed set of phrases, and includes one-way communication. A similar system is used in Iraq as a briefing aid to interrogate former Iraqi intelligence officials and to speak with civilians about information relevant to locating individuals (Chisholm, 2004). This shows the viability of the approach.

TIA-P is a commercially available system, developed by CMU, incorporating a 133 MHz 586 processor, 32MB DRAM, 2 GB IDE Disk, full-duplex sound chip, and spread spectrum radio (2Mbps, 2.4 GHz) in a ruggedized, handheld, pen-based system designed to support speech translation applications. TIA-P is shown in Fig. 15.17.

Dragon loads into memory and stays memory resident. The translation uses uncompressed ~20 KB of .WAV files per phrase. There are two channels of output: (a) the first plays in English, and (b) the second plays in Croatian. A stereo signal can be split, with one channel directed to an earphone, and the second to a speaker. This is done in hardware attached to the external speaker. An Andrea noise-canceling microphone is used with an on-off switch.

Speech translation for one language (Croatian) requires a total of 60 MB disk space. The speech recognition requires an additional 20–30 MB of disk space.



FIGURE 15.17. TIA-P Wearable Computer.

TIA-P has been tested with the Dragon speech translation system in several foreign countries: Bosnia (Fig. 15.18), Korea, and Guantanamo Bay, Cuba. TIA-P has also been used in human intelligence data collection and experimentation with the use of electronic maintenance manuals for F-16 maintenance.

The following lessons were learned during the TIA-P field tests:

- Wires should be kept to a minimum.
- Handheld display was convenient for checking the translated text.
- Standard, external electrical power should be available for use internationally.
- Battery lifetime should be extended.
- Ruggedness is important.

The smart modules (circa 1997) are a family of wearable computers dedicated to the speech processing application (Smailagic, Siewiorek, & Reilly, 2001). A smart module provides a service almost instantaneously and is configurable for different applications. The design goals also included (a) reduce latency, (b) remove context swaps, and (c) minimize weight, volume, and power consumption (Reilly, 1998; Martin, 1999). The functional prototype consists of two functionally specialized modules: (a) performing language translation and (b) speech recognition. The first module incorporates speech to text language recognition and text to speech synthesis. The second module performs text-to-text language translation. The LT module runs



FIGURE 15.18. U.S. Soldier in Balkans Using TIA-P.

the PANLITE language translation software (Frederking & Brown, 1996), and the SR module runs CMU's Sphinx II continuous, speaker-independent speech recognition software (Ravishankar, 1996; Li, Hon, Hwang, & Reddy, 1989) and Phonebox Speech Synthesis software.

Figure 15.19 depicts the structure of the speech translator, from English to a foreign language, and vice versa. The speech is input into the system through the Speech Recognition subsystem. A user wears a microphone as an input device, and background noise is eliminated using filtering procedures. A language model, generated from a variety of audio recordings and data, provides guidance for the speech recognition system by acting as a knowledge source about the language properties. The Language Translation engine uses an Example-Based Machine Translation (EBMT) system, which takes individual sentence phrases and compares them to a corpus of examples it has in memory to find phases it knows how to translate. A lexical MT (glossary) translates any unknown word that may be left. The EBMT engine translates individual "chunks" of the sentence using the source language model and then combines them with a model of the target language to ensure correct syntax. When reading from the EBMT corpus, the system makes several random-access reads while searching for the appropriate phrase. Since random reads are done multiple times, instead of loading large, continuous chunks of the corpus into memory, the disk latency times will be far more important than the disk bandwidth. The Speech Generation subsystem performs text-to-speech conversion at the output stage. To make sure that misrecognized words are corrected, a Clarification Dialog takes place on-screen. It includes the option to speak the word again, or to write it in. As indicated in Fig. 15.19, an alternative input modality could be the text from the Optical Character Recognition subsystem (such as scanned documents in a foreign language), which is fed into the Language Translation subsystem.

User interface design went through several iterations based on feedback during field tests. The emphasis was on getting completely correct two-way speech translation, and having an easy to use, straightforward interface for the clarification dialogue.

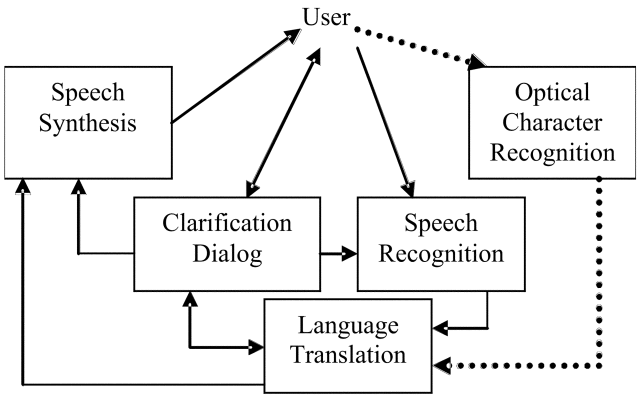


FIGURE 15.19. Speech Translator System Structure.



FIGURE 15.20. Speech Recognizer (SR) and Language Translator (LT) Smart Module.

The speech recognition code was profiled and tuned. Profiling was performed to identify "hot spots" for hardware and software acceleration, as well as to reduce the required computational and storage resources. A six-times speedup was achieved over the original desktop PC system implementation of language translation, and five times smaller memory requirements (Christakos, 1998). Reducing OS swapping and code optimization made a major impact. Input to the module is audio and output is ASCII text. The speech recognition module is augmented with speech synthesis. Figure 15.20 illustrates a combination of the language translation module (LT), and speech recognizer (SR) module, forming a complete stand-alone, audio-based, interactive dialogue system for speech translation.

Target languages included Serbo-Croatian, Korean, Creole French, and Arabic. Average language translation performance was one second per sentence.

The key factors that determine how many processes can be run on a module are memory, storage space, and available CPU cycles. To minimize latency, the entirety of an application's working dataset should be able to stay memory resident.

Figure 15.21 depicts the functional prototype of the Speech Translator Smart Module, with one module performing language translation, and another one speech recognition and synthesis.

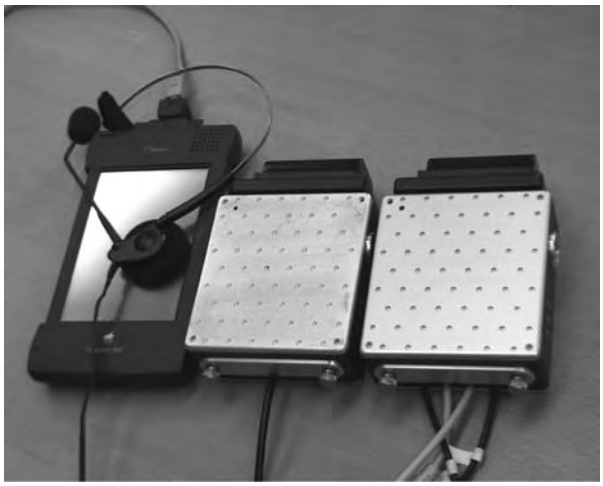


FIGURE 15.21. Speech Translator SM Functional Prototype.

PERFORMANCE EVALUATION

Figure 15.22 illustrates the response time for speech recognition applications running on TIA-P and SR Smart Module. As SR is using a lightweight operating system (Linux) versus Windows 95 on TIA-P and the speech recognition code is more customized, it has a shorter response time. An efficient mapping of the speech recognition application onto the SR Smart Module architecture provided a response time very close to real-time. To ensure system responsiveness, it was important to provide feedback to the person in near real time.

The lessons learned from tests and demonstrations include: manual intervention process to correct misrecognized words incurs some delay; swapping can diminish the performance of the language translation module; the size of display can be as small as a deck of cards.

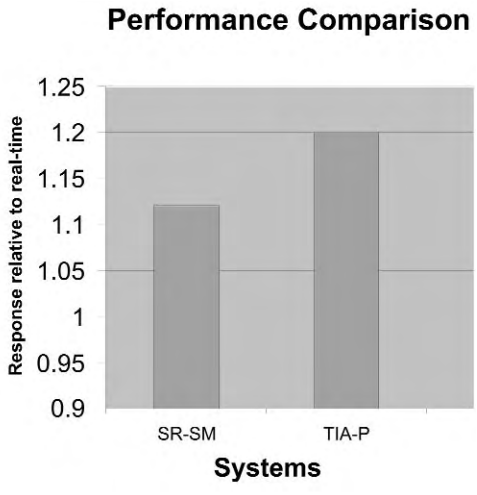


FIGURE 15.22. Response Times (lower is better).

The required system resources for speech translator software are several times smaller than for the laptop/workstation version, as shown in Table 15.2.

Dual Purpose Speech

In industry, most speech recognition on mobile computers concentrates on the tasks of form filling or simple interface commands and navigation. One reason is that speech interfaces are often socially interruptive when other people are nearby. Speech translation, as with the TIA system above, is a different class of interface. The computer is an essential enabler of the conversation. Lyons, Skeels, and colleagues (2004) introduced a different type of conversation enabler in their Dual Purpose Speech work.

Dual Purpose Speech is easiest to discuss using a scenario. Tracy, a wearable user equipped with a head-up display (HUD) and a Twidder keyboard, is in conversation with a recently introduced colleague. Pressing a button on the keyboard, the wearable user enables speech recognition and says, "Bob, what is your phone number so that I have it for later?"

The wearable recognizes that its user wants to record a phone number and starts the user's contact list application. It attempts to recognize the name spoken and enters that into the application; however, it also saves the speech so that the user can correct the text later if there is an error.

Bob responds, "Area code 404"
"404," repeats Tracy.
"555-1212," completes Bob.
"555-1212," continues Tracy, who presses another button on her keyboard indicating the interaction is over. "Okay, I have it!"

On Tracy's head-up display a new contact has been made for "Bob (404) 555-1212." When Tracy finishes her conversation, she clicks an "Accept" button on the application because she has recognized the information correctly. Tracy could also edit the information or play back the audio recorded during the interaction with Bob. Note that Tracy verbally repeated the information that Bob provided—a good conversational practice. Tracy both confirmed that she understood the information and provided Bob with an opportunity to correct her if necessary; however, this practice is also good from a privacy standpoint. Tracy wears a noise-canceling microphone, which is thresholded to record only her own voice and not that of her conversational partners. In this way, Tracy respects the privacy of her colleagues.

Lyons, Skeels, and colleagues (2004) have designed Dual Purpose Speech applications for scheduling appointments, providing reminders for the user, and communicating important information to close colleague. The key point of this research from the perspective of this section, however, is that these applications allow the user to manipulate information on their wearables as part of the process of communicating (thus, the "dual purpose" name). The users may actively format their speech so the system can better understand them, and they may have to correct the system afterwards; however, the interface is manipulated and the information is entered as part of a social process.

This style of interface provides a contrast to the traditional desktop computer, where the user's attention is assumed to be dedicated to the interface. Other wearable-computing-related fields also attempt to create interfaces that are driven by the user's interactions with the environment. For example, Feiner, MacIntyre, and Seligmann (1993) early augmented reality systems attempted to display appropriate repair instructions based on the user's actions during the repair process. Such awareness of the user's context and goals may allow wearable computers to be utilized where a user's lack of attentional or physical resources would normally preclude traditional desktop applications.

TABLE 15.2. Comparison of Required System Resources

	Laptop/ Workstation	Functional Module SR/LT	Optimized Module SR/LT
Memory Size	195 MB	53 MB	41 MB
Disk Space	1 GB	350 MB	200 MB

Perception

Just as dexterity is impaired when a user is on the go, the user's ability to perceive a wearable's interface is also lessened. The vibration and visual interference from a moving background interferes with visual tasks. Background noise and the noise from the body itself affect hearing. The moving of clothes over the body and the coupling of mechanical shock through the body can lessen the user's ability to perceive tactile displays. Sears, Lin, Jacko, and Xiao (2003) described these detriments to mobile interaction caused by environmental and situational factors as "Situationally-Induced Impairments and Disabilities". These researchers and others are developing procedures to test human performance in mobile computing tasks in context (in this case, walking a path; Barnard, 2005, in press). Such research is sorely needed, as not enough is known about how to adequately simulate mobile computing scenarios in testing. For example, in Barnard and colleagues' (in press) work on performing reading comprehension tasks on PDAs while walking, lighting levels affected workload measures more when walking a path than when walking on a treadmill. The community needs to develop understanding about the interactions between mobility, attention, and perception in common mobile computing scenarios in order to adequately develop testing environments for mobile interfaces.

In the past, such work focused on cockpits, both for aviation and automobiles (Wickens, C. D. & Hollands J. (1999). Engineering Psychology and Human Performance (3rd Edition). Prentice Hall; Melzer, & Moffit, 1997; Velger, 1998); however, the U.S. military's Land Warrior project has highlighted the need for such research for dismounted users who are on the go (Blackwood, 1997). Some researchers have begun exploring mobile output devices for very specific tasks. For example, Krum (2004) described experiments with a head-up display that focused on determining how to render overhead views of an area to encourage learning of the layout of the surrounding environment while the user is navigating to a goal on foot. As mobile-augmented reality is becoming practical from a technical standpoint, researchers have begun to address perceptual issues. While not a mobile experiment, Laramee and Ware (2002) have investi-

gated head-mounted displays to determine the relative effects of rivalry and visual interference between binocular and monocular displays with varying levels of transparency. As the market determines which mobile contexts are most important for users, experiments such as these will help determine how to design interfaces to least interfere with the user's primary tasks while providing the most value in terms of augmentation.

CONCLUSION AND FUTURE CHALLENGES

Wearable computers are an attractive way to deliver a ubiquitous computing system's interface to a user, especially in non-office-building environments. The biggest challenges in this area deal with fitting the computer to the human in terms of interface, cognitive model, contextual awareness, and adaptation to tasks being performed. These challenges include

- User interface models. What is the appropriate set of metaphors for providing mobile access to information (e.g., what is the next "desktop" or "spreadsheet")? These metaphors typically take over a decade to develop (e.g., the desktop metaphor started in early 1970s at Xerox PARC and required over a decade before it was widely available to consumers). Extensive experimentation working with end-user applications will be required. Furthermore, a set of metaphors may each be tailored to a specific application or a specific information type.
- Input/output modalities. While several modalities mimicking the input/output capabilities of the human brain have been the subject of computer science research for decades, the accuracy and ease of use (e.g., many current modalities require extensive training periods) are not yet acceptable. Inaccuracies produce user frustrations. In addition, most of these modalities require extensive computing resources that will not be available in low-weight, low-energy, wearable computers. There is room for new, easy-to-use input devices such as the dial developed at Carnegie Mellon University for list-oriented applications.
- Quick interface evaluation methodology. Current approaches to evaluate a human computer interface requires elaborate procedures with scores of subjects. Such an evaluation may take months and is not appropriate for use during interface design. These evaluation techniques should especially focus on decreasing human errors and frustration.
- Matched capability with applications. The current thought is that technology should provide the highest performance capability; however, this capability is often unnecessary to complete an application and enhancements such as full-color graphics require substantial resources and may actually decrease ease of use by generating information overload for the user. Interface design and evaluation should focus on the most effective means for information access and resist the temptation to provide extra capabilities simply because they are available.
- Context-aware applications. How do we develop social and cognitive models of applications? How do we integrate input from multiple sensors and map them into user social and cognitive states? How do we anticipate user needs? How do we interact with the user? These, plus many other questions, have to be addressed before context aware computing becomes possible. Some initial results have been reported in Krause, Smailagic, and Siewiorek (2006).

References

Anhalt, J., Smailagic, A., Siewiorek, D., et al. (2001). Towards context aware computing. *IEEE Intelligent Systems, 6*(3), 38–46.

Azuma, R. (1997). A survey of augmented reality. *Presence, 6*(4), 355–386.

Barnard, L., Yi, J. S., Jacko, J. A., & Sears, A. (2005). An Empirical Comparison of Use-in Motion Evaluation Scenarios for Mobile Computing Devices. *International Journal of Human Computer Studies, 62*, 487–520.

Barnard, L., Yi, J., Jacko, J. A., & Sears, A. (in press). Capturing the Effects of Context on Human Performance in Mobile Computing Systems. *Personal and Ubiquitous Computing*.

Blackwood, W. (1997). *Tactical display for soldiers*. Washington, DC: National Academy of Sciences. pp. 57–61

Chisholm, M. (2004). Technology that speaks in tongues. *Military Information Technology, 8*(2). Retrieved from http://www.military-information-technology.com/article.cfm?DocID=424

Christakos, C. K. (1998). *Optimizing a language translation application for mobile use*. Unpublished master's thesis, Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, PA.

Clawson, J., Lyons, K., Starner, T., & Clarkson, E. (2005). The impacts of limited visual feedback on mobile text entry using the mini-QWERTY and Twiddler keyboards. *Proceedings of the IEEE International Symposium on Wearable Computers*. Los Alamitos, CA: IEEE Computer Society Press.

Feiner, S., MacIntyre B., & Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM, 36*(7), 52–62.

Frederking, R. E., & Brown, R. (1996). The Pangloss-lite machine translation system: Expanding MT horizons. *Proceedings of the Second Conference of the Association for Machine Translation in the Americas* (pp. 268–272). Montreal, Canada.

Gemperle, F., Kasabach, C., Stivoric, J., Bauer, B., & Martin, R. (1998). Design for wearability. *Second International Symposium on Wearable Computers* (pp. 116–122). Los Alamitos, CA: IEEE Computer Society Press.

Iachello, G., Smith, I., Consolvo, S., Abowd, G., Hughes, J., Howard, J., et al., (2005). Control, deception, and communication: evaluating the deployment of a location-enhanced messaging service. *Proceedings of the Seventh International Conference on Ubiquitous Computing* (pp. 213–231). Los Alamitos, CA: IEEE Computer Society Press.

Kollmorgen, G. S., Schmorrow. D., Kruse, A., & Patrey, J. (2005). The cognitive cockpit-state of the art human-system integration. *Proceedings of the November 2005 Interservice/Interindustry Training Simulation and Education Conference*, DARPA, Arlington, VA.

Krause, A., Smailagic, A., & Siewiorek, D. P. (2006). Context-aware mobile computing: learning context-dependent personal preferences from a wearable sensor array. *IEEE Transactions on Mobile Computing, 5*(2), 113–127.

Krum, D. (2004). *Wearable computers and spatial cognition.* Unpublished doctoral dissertation, Georgia Institute of Technology, College of Computing, Atlanta.

Lee, K.-F., Hon, H.-W., Hwang, M.-Y., Mahajan, S., Reddy, R. (1989). *International Conference Acoustics, Speech, and Signal Processing* (ICASSP'89) May 23–26, (vol. 1, pp. 445–448). IEEE Computer Society Press, Los Alamitos, CA.

Laramee, R., & Ware, C. (2002). Rivalry and interference with a head mounted display. *ACM Transactions on Computer-Human Interface, 9*(3), 238–251. New York: ACM Press

Lyons, K., Skeels, C., Starner T., Snoeck, B., Wong, B., & Ashbrook, D. (2004). Augmenting conversations using dual-purpose speech. *Proceedings, User Interface and Software Technology* (pp. 237–246). Los Alamitos, CA:

Lyons, K., Starner, T., & Plaisted, D. (2004) Expert typing using the Twiddler one handed chord keyboard. *Proceedings of the IEEE International Symposium on Wearable Computers* (pp. 94–101).

Lyons, K., & Starner, T. (2001). Mobile capture for wearable computer usability testing. *Proceedings of the International Symposium on Wearable Computers*.

Martin, T. (1999). *Balancing batteries, power, and performance: System issues in CPU speed-setting for mobile computing.* Unpublished doctoral dissertation, Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, PA.

Melzer, J., & Moffitt, K. (1997). *Head mounted displays: Designing for the user. New York:* McGraw-Hill.

Ockerman, J. (2000). Task guidance and procedure context: Aiding workers in appropriate procedure following. Unpublished doctoral dissertation, Georgia Institute of Technology, School of Industrial System Engineering, Atlanta. New York: ACM Press

Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahit, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (pp. 919–928).

Ravishankar, M. (1996). *Efficient algorithms for speech recognition*. Unpublished doctoral dissertation, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA.

Reilly, D. (1998). *Power consumption and performance of a wearable computing system*. Unpublished master's thesis, Carnegie Mellon University, Electrical and Computer Engineering Department, Pittsburgh, PA. New York: Springer

Sears, A., Lin, M., Jacko, J., & Xiao, Y. (2003). When Computers Fade: Pervasive Computing and Situationally-Induced Impairments and Disabilities. *Proceedings of HCII 2003* (pp. 1298–1302).

Siewiorek, D. P. (2002). Issues and challenges in ubiquitous computing: new frontiers of application design. *Communications of the ACM, 45*(12), 79–82.

Siewiorek, D. P., Smailagic, A., & Lee, J. C. (1994). An interdisciplinary concurrent design methodology as applied to the Navigator wearable computer system. *Journal of Computer and Software Engineering*, *2*(2), 259–292.

Smailagic, A. (1997). ISAAC: A voice activated speech response system for wearable computers. *Proceedings of the IEEE International Conference on Wearable Computers*. IEEE Computer Society Press, Los Alamitos, CA.

Smailagic A., Siewiorek D. P., Martin, R., & Stivoric, J. (1998). Very rapid prototyping of wearable computers: A case study of VuMan 3 custom versus off-the-shelf design methodologies. *Journal of Design Automation for Embedded Systems, 3*(2–3), 219–232. IEEE Computer Society Press, Los Alamitos, CA.

Smailagic, A., & Siewiorek, D. P. (1994). The CMU mobile computers: A new generation of computer systems. *Proceedings of the IEEE COMPCON 94* (pp. 467–473). Los Alamitos, CA: IEEE Computer Society Press.

Smailagic, A., & Siewiorek, D. (1996). Modalities of interaction with CMU wearable computers. *IEEE Personal Communications, 3*(1), 14–25. IEEE Computer Society Press, Los Alamitos, CA.

Smailagic, A., Siewiorek, D. P., & Reilly, D. (2001). CMU wearable computers for real-time speech translation. *IEEE Personal Communications, 8*(2), 6–12. New York: ACM Press

Smailagic, A., & Siewiorek, D. (1993). A case study in embedded system design: the VuMan 2 wearable computer. *IEEE Design and Test of Computers, 10*(3), 56–67.

Starner, T., Snoeck, C., Wong, B., & McGuire, R. (2004) Use of mobile appointment scheduling devices. *Proceedings of the ACM Conference Human Factors in Computing Systems* (pp. 1501–1504). Los Alamitos, CA: IEEE Computer Society Press.

Starner, T. (2001). The challenges of wearable computing: Part 1+2. *IEEE Micro*, *21*(4), 44–67.

Stein, R., Ferrero, S., Hetfield, M., Quinn, A., & Krichever, M. (1998). Development of a commercially successful wearable data collection system. *Proceedings of the IEEE International Symposium on Wearable Computers* (pp. 18–24).

Velger, M. (1998). *Helmet-mounted displays and sights*. Norwood, MA: Artech House, Inc.

Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman & R. Davies (Eds), *Varieties of attention.* New York: Academic Press.

Wickens, C. D., & Hollands, J. (1999). *Engineering psychology and human performance* (3rd edition). Prentice Hall.

·16·

DESIGN OF COMPUTER WORKSTATIONS

Michael J. Smith and Pascale Carayon
University of Wisconsin–Madison

William J. Cohen
Exponent, Inc.

Ergonomic Perspective **314**
Studies on the Design of Fixed Workstations **315**
Guidelines for Fixed Computer Workstation Design Features **316**
The Design of Workstations for Use with Laptop Computers **317**
PIDs, Carry Along, and Wearable IT **318**
The Chair as a Critical Element of the Workstation . . . **319**
Additional Workstation Considerations **321**
The Visual Environment **321**
The Auditory Environment **322**
Ergonomic Implementation Issues **323**
Summary of Recommendations **324**
References **324**

In our last version of this chapter, we focused on fixed location computing workstations and situations, but acknowledged the burgeoning use of portable information devices (PIDs) and provided a few ideas to consider about their use. We indicated that fixed computer environments, while still a significant part of workstation design considerations, have been surpassed by highly mobile information technology that does not use fixed workstations. These portable technologies are now in use in almost every venue and human activity, and the nature of their characteristics and activities of use do not lend them to traditional fixed workstation considerations. This introduces a host of potential ergonomic concerns related to the design of work areas (and activities) in which PIDs and other forms of computing are used. Decades of research and applications have defined important considerations in the ergonomic design of fixed computer work areas (Grandjean, 1979, 1984; Stammerjohn, Smith, & Cohen, 1981; Cakir, Hart, & Stewart, 1979; ANSI/HFES-100, 1988; Smith & Cohen, 1997; Smith, Carayon, & Cohen, 2003; BSR/HFES-100, 2005). However, very little has been done to define the design of work areas for PIDs and mobile computing. In this chapter, we will propose some ideas and considerations for dealing with ergonomic concerns for these mobile technologies in addition to updating information on fixed computer workstation applications. However, we admit that the challenge to define good ergonomic practice for the use of PIDs is large, and our advice is still very limited.

Ergonomics is the science of fitting the environment and activities to the capabilities, dimensions, and needs of people. Ergonomic knowledge and principles are applied to adapt working conditions to the physical, psychological, and social nature of the person. The goal of ergonomics is to improve performance while enhancing comfort, health, and safety. Computer workstation design is more than just making the computer interfaces easier to use, or making furniture adjustable in various dimensions. It also involves integrating design considerations with the work environment, the task requirements, the social aspects of work, and job design. Critical considerations for good ergonomics practice are to limit biomechanical loading on the back and joints to low forces of short duration; to keep the back, neck, and joints in good postures while working; to limit the amount of time you are in static postures; to limit the duration and extent of repetitive motions of any part of your body including your voice; to work in environments where you can easily see and hear; and to take frequent rest breaks from working.

A fundamental perspective in this chapter is that work area (workstation) design influences employee comfort, health, motivation, and performance. We will examine basic ergonomic considerations (principles, practices, concerns) that can be used to develop guidance for the design of work areas (workstations) for the use of computing and related IT such as PIDs. Some of today's technologies have the capability to interact directly with each other, sometimes without human intervention, while other times requiring human action. The wide myriad of environments of use, interaction schemes, and activities of use make specific guidance for PIDs complex and difficult. Thus, we will provide general guidance to reduce musculoskeletal stress and to enhance physical comfort.

ERGONOMIC PERSPECTIVE

Forty years ago, a person would interact with several different types of information sources and technologies when engaging in activities, such as work. The person might look at hardcopy documents, take notes with a pen and a paper tablet, use a fixed location telephone, talk face-to-face with colleagues, type on a typewriter, and perform many other tasks during the course of the working day. The diversity of activities led people to move actively around during the day. Then, thirty years ago, many people started to spend most of their workdays in sedentary work sitting in front of a computer terminal. This type of human-machine system led to restricted physical movement and employee attention directed toward the computer monitor. Today there are mixed exposures in a multidimensional environment where some people are still at fixed sedentary workstations, while others are interacting with multiple technologies that have different physical characteristics. In some instances, people are again on the move and not in fixed, sedentary situations. Movement is good for the muscles while sedentary sitting is not if it lasts for too long. However, the interaction with technology while moving may create dynamic postural loading and the awkward application of perceptual-motor skills when making inputs to technology or when reading and listening to displays. For example, you might be walking through the airport while checking your email on your cell phone.

In Sweden, since the initial work of Hultgren and Knave (1973), Ostberg (1975), and Gunnarsson and Ostberg (1977), thousands of research studies from every corner of the globe have examined the working conditions of computer users and their associated health complaints. There have been several international conferences devoted to these issues starting in 1980, and conferences on these issues are already programmed through 2010. The findings from this research and the meetings have generally indicated that poor ergonomic and task design conditions are associated with large numbers of computer users complaining about visual discomfort, musculoskeletal discomfort and pain, and psychological distress. In fact, if these adverse ergonomic conditions are repeated daily over a long period, more or less chronic aches and pains of the upper extremities and back can occur, and may involve not only muscles but also other soft tissues, such as tendons and nerves. Long lasting, adverse ergonomic conditions may lead to a deterioration of joints, ligaments, and tendons. Reviews of field studies for fixed computer workstations (Bergquist, 1984; Bergquist et al., 1992; Smith, 1984, 1987, 1997; Grandjean, 1979, 1987; Hagberg et al., 1995; Bernard, 1997; Carayon, Smith, & Haims, 1999), as well as general experience, have shown that these conditions may be associated with a higher risk of

1. Inflammation of the joints
2. Inflammation of the tendon-sheaths
3. Inflammation of the attachment-points of tendons
4. Symptoms of degeneration of the joints in the form of chronic arthroses
5. Painful muscles
6. Disc troubles
7. Peripheral nerve disorders

The relationship between the user and technology is reciprocal. It is a system where the constraints of one element affect the performance of the other. Various aspects of the system, such as the task requirements, the work demands, the environment, and the workstation, all influence how effectively and comfortably the technology can be used by the person (Smith & Sainfort, 1989; Smith & Carayon, 1995; Carayon & Smith, 2000). One of the consequences of this is that the design of the workspace limits the nature and effectiveness of the interaction between the person and the technology. For instance, inadequate space for carrying out physical activities can lead to constrained postures, which together with long lasting static loading produces discomfort in the back and shoulders. This leads to symptoms of tiredness, muscle cramps, and pain. In addition, heavy workload, chronic repetition, and other biomechanical strains can cause similar problems. These adverse postural, repetition, and workload exposures lead to reduced performance and productivity, and in the long run, they may also affect employee well being and health.

Today, people carry their computing and communications with them and engage in activities in any available work area. The potential for serious ergonomic risks is high for these new modes of using technology. We will first examine problems and design considerations for fixed workstations where PCs and laptops are used. Then we will look at PIDs.

STUDIES ON THE DESIGN OF FIXED WORKSTATIONS

Workstation design is a major element in ergonomic strategies for improving user comfort and particularly for reducing musculoskeletal problems. Often, the task requirements will have a role in defining the layout and dimensional characteristics of the workstation. The relative importance of the screen, input devices, and hardcopy (e.g., source documents) depends primarily on the task, and this can influence the design considerations necessary to improve operator performance, comfort, and health. Many studies have examined fixed computer workstation design issues and generally have found that the quality of the fit between the user and the interactive devices being used has a substantial role in user musculoskeletal comfort and musculoskeletal pain (Smith, Carayon, & Cohen, 2003). For instance, Cohen, James, Taveira, Karsh, Scholz, and Smith (1995) identified the following working conditions that led to awkward posture and undue loads on the musculoskeletal system:

1. Static postures of the trunk neck and arms
2. Awkward twisting and reaching motions
3. Poor lighting and glare
4. The placement of the keyboard on uneven working surfaces
5. Insufficient work surface space
6. Insufficient knee and toe space
7. The inability for the chair armrests to fit under the working surfaces
8. Chairs with poor back and shoulder support, inadequate padding in the backrest and seat pan, arm rests that did not fit under working surfaces, and a lack of appropriate seat pan height adjustment

Based on this and several other studies, ergonomics experts have proposed guidance for designing fixed computing workstations. Grandjean (1984) proposed the consideration of the following features of fixed computer workstation design:

1. The furniture should be as flexible as possible with adjustment ranges to accommodate the anthropometric diversity of the users.
2. Controls for workstation adjustment should be easy to use.
3. There should be sufficient knee space for seated operators.
4. The chair should have an elongated backrest with an adjustable inclination and a lumbar support.
5. The keyboard should be moveable on the desk surface. (Today we might say the input devices should be able to be positioned at several locations on the working surfaces.)

Derjani-Bayeh and Smith (1999; Smith & Derjani-Bayeh, 2003) conducted a prospective intervention study to examine the benefits of ergonomic redesign for computer users at fixed workstations. The study took place in a consumer products call center where shoppers could order products from a catalog using a telephone or an Internet site. There were three ergonomic interventions studied. In the first condition, ergonomics experts provided modifications to current workstation configurations to maximize their fit with the incumbent employee. In the second condition, new workstation accessories (keyboard tray, monitor holder, document holder, wrist rest, footrest, and task lighting) were added as needed to improve the employees' fit with their workstations and general environment. In the third condition, the same factors as the second condition were added but in addition, a new chair with multiple adjustments was also added.

Eighty volunteer subjects participated. They were drawn from a larger pool of volunteers. The participants for the third condition were randomly selected from the larger pool, and then subjects for conditions one and two were matched to these selections based on type of job, age, gender, and length of experience with the company. Baseline measurements of self-reported health status were collected using a questionnaire survey. Follow-up measurements were taken directly after implementation of the ergonomic improvements and then 12 months later. In addition, productivity measurements were obtained for each participant and a control group of approximately 375 employees in the same departments. The results indicated that subjects working under conditions two and three showed reductions in the extent and intensity of musculoskeletal health complaints, but not the subjects in condition one. However, the subjects in condition one showed greater average improvement in productivity than in conditions two and three and to the control group receiving no treatment. Of importance to designers was the finding that not all subjects showed improved productivity with ergonomic improvements. In fact about one-half of the subjects showed reduced productivity with the ergonomic improvements, even though the overall average for the ergonomic improvements showed a positive effect.

GUIDELINES FOR FIXED COMPUTER WORKSTATION DESIGN FEATURES

Based on the findings from several field studies and standards dealing with computer workstation design (ANSI/HFES-100, 1988; CSA, 1989; BSR/HFES-100, 2005) the following general guidance is proposed for the design of fixed computer workstations.

The recommended size of the work surface is dependent upon the tasks, the documents, and the technologies being used. The primary working surface (e.g., those supporting the keyboard, the mouse, the displays, and documents) should be sufficient to:

1. Allow the display screens to be moved forward or backward for a comfortable viewing distance for different employees with differing visual capabilities.
2. Allow a detachable keyboard and a detachable mouse (or other pointing device) to be placed in several locations on the working surface.
3. Allow source documents to be positioned for easy viewing and proper musculoskeletal alignment of the upper extremities and the back.
4. Additional working surfaces (e.g., secondary working surfaces) may be necessary to store, layout, read, write on, or manipulate documents, materials, or technologies (input devices, displays, computers, PIDs).
5. Provide adequate knee and legroom for repositioning movements while working.
6. The tabletop should be as thin as possible for better thigh and knee clearance.
7. Establish a comfortable table height that provides the necessary thigh and knee clearance, and which allows input devices (keyboard, mouse) to be at comfortable heights. Adjustable tables provide the opportunity to better fit the users.

Sometimes computer workstations are configured so that multiple pieces of equipment and source materials can be equally accessible to the user. In this case, additional working surfaces are necessary to support these additional tools and should be arranged to allow for easy movement from one surface to another. Proper clearances under each working surface should be maintained, as well as a comfortable height.

It is important to provide unobstructed room under the working surface for the feet and legs so that operators can easily shift their posture. Knee space height and width and toe depth are the three key factors for the design of clearance space under the working surfaces. The recommended minimum width for leg clearance is 51 cm, while the preferred minimum width is 61 cm (ANSI/HFS-100, 1988). The BSR/HFES-100 (2005) guidance calls for minimum width of leg clearance of 50 cm. The minimum depth under the work surface from the operator edge of the work surface should be 38 cm for clearance at the knee level and 59 cm at the toe level (ANSI/HFS-100, 1988). The BSR/HFES-100 (2005) recommends a minimum depth at the extended leg toe level of 60 cm. A good workstation design accounts for individual body sizes and often exceeds minimum clearances to allow for free postural movement.

Table height has been shown to be an important contributor to computer user musculoskeletal problems (Hünting, Läubli, & Grandjean, 1981; Grandjean, Nishiyama, Hünting & Pidermann, 1982; Grandjean, Hünting & Nishiyama, 1982, 1984; Grandjean, Hünting, & Pidermann, 1983). Normal desk height of 30 inches (76 cm) is often too high for keyboard and mouse use by most people. It is desirable for table heights to vary with the height of the user, particularly if the chair is not height adjustable. Height adjustable working surfaces are effective for this. Adjustable multisurface tables encourage good posture by allowing the keyboard, mouse, and displays to be independently adjusted to appropriate keying, pointing, and viewing heights for each user and each task. Tables that cannot be adjusted easily can be a problem when used by multiple users of differing sizes, especially if the chair is not height adjustable. When adjustable tables are used, the ease of making the adjustments is essential. Adjustments should be easy to make, and operators should be instructed on how to adjust the workstation to be comfortable.

Specifications for seated working surfaces' heights vary with whether the table is adjustable or at one fixed height, and with a single working surface or multiple working surfaces. The proper height for a nonadjustable working surface is about 70 cm (27.5 inches) from the floor to top of the working surface. However, it must be emphasized that fixed height table surfaces fit only a portion of the user population. In the absence of a height adjustable chair, such fixed height table surfaces will produce a misfit in a portion of the users that could influence musculoskeletal comfort and pain.

The BSR/HFES-100 (2005) guidance requires that all surfaces supporting input devices must be height adjustable (or a combination of height and tilt adjustable). In addition, the input-device support surfaces should provide for adjustment fore and aft and side-to-side if possible.

Adjustable tables allow vertical adjustments of the keyboard (and mouse) and displays. Some allow for the independent adjustment of the keyboard and display. For a single adjustable seated working surface, the working-surface height adjustment range should be between 60 cm to 80 cm (23.5–31.5 inches). The BSR/HFES-100 (2005) guidance is for an adjustment range of 56 cm to 72 cm (22.0–28.5 inches). In the case of independently adjustable working surfaces for the keyboard and the screen, the appropriate height ranges are 59 cm to 71 cm (23.0–28.0 inches) for the keyboard surface (ANSI/HFS-100, 1988), and 90 cm to 115 cm (35.5–45 inches) for the screen surface (Grandjean, 1987).

The BSR/HFES-100 (2005) guidance indicates that the worktable surfaces should provide for tilt from −15 degrees to +20 degrees if possible.

For standing only workstations, the BSR/HFES-100 (2005) recommends a table surface height adjustment range of 95 cm to 118 cm (37.5–46.5 inches). However, if both height and tilt surface adjustments are included, then the range of height adjustment recommended is from 78 cm to 118 cm (30.5–46.5 inches) with tilt adjustment ranges from −45 degrees to +20 degree. For sit and stand workstations, the BSR/HFES-100 recommendation for height adjustment of the input device surfaces is between 56 cm to 118 cm (22.0–46.5 inches).

THE DESIGN OF WORKSTATIONS FOR USE WITH LAPTOP COMPUTERS

We will start with the laptop as a prime example of the influence of portability and efficiency, and then we will move on to other portable IT devices. Recently the Human-Computer Interaction Committee of the International Ergonomics Association (IEA) produced a guideline for the use of laptop computers to improve ergonomic conditions (Saito et al., 2000). This was prompted by the ever-increasing sales of laptops and the replacement of fixed PCs by portable laptops on fixed working surfaces.

The primary advantage of the laptop is easy portability so the user can take the computer anywhere to do work. He or she can use it at the office, take it home to finish work, or take it with him or her to make a presentation at a meeting. The convenience and effectiveness of easy, lightweight portability are very high. In addition, all of the files are with the laptop, so nothing is mistakenly left behind at the office (or home). However, the comfort and health factors can be very low if the person uses the laptop in all manner of environments, workstations, and tasks that diminish the consistent application of good ergonomic principles. An important feature of the IEA laptop guideline (Saito, 2000) is to encourage situations of use that mirror the best practices of ergonomic conditions for fixed computer workstations.

Work Environment and Workstation Layout, "Create an environment that fits your work"

1. Use your laptop in a proper environment (lighting, temperature, noise, and so on). In particular, make sure the work area is neither too bright nor too dark.
2. Allocate enough space on your desk when placing a laptop.

Chair and Desk, "Adjust chair height to match your physique"

1. Adjust your chair height based on the height of the keyboard, such that your forearm is parallel to the surface of the keyboard.
2. If your feet do not lie flat on the floor, provide a footrest.
3. Provide enough space underneath the desk.

Keyboard, "Set the keyboard to a desirable angle, and use a palm rest if necessary"

1. Adjust the angle of the keyboard based on your posture and preferences.
2. Make sure there is space in front of the keyboard for you to rest your wrists comfortably (this space can be on the desktop surface itself if the keyboard is thin).
3. If the keyboard seems difficult to use, use an external keyboard.

Working Posture, "Avoid unnatural postures, and change your posture occasionally"

1. Avoid staying in postures where you are bent too far forward or backward, or twisted, for an extended duration
2. Laptop users tend to view the display from too close, so make sure you maintain a distance of at least 40–50 cm between the display and your eyes.
3. Alternate near vision with far vision (i.e., observe object located at least 6 m far) as frequent as possible.
4. Make sure your wrists are not at an unnatural angle.

Nonkeyboard Input Devices, "Use a mouse as your pointing device if at all possible"

1. If a mouse can be connected to your laptop, then do so as often as possible.Use a mouse pad whenever you use a mouse
2. When you cannot connect a mouse, make sure you understand the built-in pointing device, and use the pointing device appropriately (Saito et al., 2000, pp. 421–434)

These laptop guidelines provided by Saito et al. (2000) are useful when the laptop is used as a fixed PC at a docking station or at a desk (worktable). However, they do not provide as much help in the situations where there is no fixed workstation. We will describe some situations below where this can occur.

Imagine sitting at the airport, and your flight has been delayed for two hours. Your have your laptop with you, so you decide the get some work done while you wait. You could rent a cubicle or kiosk at the airport that would provide you with a high-speed Internet connection, a stationary telephone, a working surface (desk or table), a height adjustable chair, and some privacy (noise control, personal space). The characteristics of these work areas do not often conform to the best principles of ergonomic design. It is likely that the cubicle will provide some improvement over sitting with the laptop on your lap, but the characteristics may not meet the recommendations presented in this chapter. Such situations are acceptable for short exposures of up to 60 minutes, but longer exposures may lead to musculoskeletal discomfort, pain, and injury (if chronic). Now imagine you have been told to stay in the boarding area because the departure may be sooner than two hours. You get out your laptop, connect it to your cell phone, and place them on your lap. (That is why they are called laptops.) You are sitting in a nonadjustable chair with poor back support.

This scenario is all too common. You can walk through O'Hare International Airport on any given day and see hundreds of people sitting at the boarding gate working on laptops that are sitting on their laps. Now imagine a palm held device that allows you to access your e-mail or to connect to the Internet. This device can be operated while you are standing in line at the airport to check in, or sitting at the boarding gate like the laptop users. You can stand or sit punching at miniature buttons (sometimes with a stylus because they are so small) and interact with the interconnected world. Again, this scene is all too familiar in almost any venue (e.g., airport, restaurant, street, office).

In situations where there is not a fixed workstation, the device is typically positioned wherever it is convenient. Very often such positioning creates bad postures for the legs, back, shoulders, arms, wrists/hands, or neck. In addition, the smaller dimensions of the manual input devices (touch pad, buttons, keyboard, joystick, roller ball) make motions much more difficult, and these often produce constrained postures and/or the use of too much force to operate the device. If the devices are used continuously for a prolonged period (such as one hour or more), muscle tension builds up and discomfort in joints, muscles, ligaments, tendons, and nerves can occur. Some devices use voice and audio interfaces and can be used when you are walking (or even running). These might be headsets with earplugs and a microphone. These devices put additional load on the neck, and the voice and audio interfaces can strain the voice or the ears.

We know very little about the discomfort, health effects, and psychosocial effects of these new strains. However, it is intuitive that frequent use of these interfaces will lead to strain and potential adverse consequences.

To reduce the undesirable effects of the poor workstation characteristics that lead to the discomfort the following recommendations are given:

1. If you are using a laptop on your lap, find a work area where you can put the laptop on a table (rather than on your lap). Then arrange the work area as closely as possible with the recommendations presented above in the IEA laptop guidelines (Saito, 2000).
2. If you are using a handheld PID, you should position yourself so that your back is supported. It is preferable to use the device sitting down. Of course, if you are using the PID as you are walking, then this is not possible. If the PID has a voice interface, then use an earpiece and a microphone so that you do not have to hold it with your hand. Be sure not to overuse the device such that your voice and ears are strained. Take frequent breaks from use to provide recovery for your voice and ears.
3. Never work in poor postural conditions for more than 30 minutes continuously. Take at least a 5-minute break (preferably 10 minutes) away from the laptop/PID use, put the device down (away), get up and stretch for 1 minute or more, and then walk for 2–3 minutes (unless you are already walking in which case you should sit down for 5 minutes). If you are using a handheld PID in a standing position, then during your break put it away, do 1 minute of stretching, and then sit down for 4 minutes. That may mean sitting on the floor, but preferably, you will sit where you can support your back (against a wall, or a seat back).
4. Buy equipment that provides the best possible input interfaces and displays (screens, headphones, typing pads). Since these devices are small, the perceptual motor requirements for their use are much more difficult (sensory requirements, motion patterns, skill requirements, postural demands, and force demands). Therefore, screens should provide easily readable characters (large, understandable), and input buttons should be easy to operate (large, properly spaced, easily accessible, low force).
5. Only use these devices when you do not have access to fixed workstations that have better ergonomic characteristics. Do not use these devices continuously for more than 30 minutes.

PIDS, CARRY ALONG, AND WEARABLE IT

We have already discussed some ergonomic issues of using PIDs. We can foresee an even greater potential for ergonomic concerns with the use of PIDs than with laptop computers. PIDs are made to be as small and light as possible for portability and convenience of carrying. This is good as the lower weight produces smaller loads of the body. Small sized devices are more difficult to manipulate with the hands, and to observe displays with the eyes. This has led many designers to emphasize verbal, auditory, and haptic interfaces that do not require substantial manipulation by the hands or good vision (Hirose & Hirota, 2005; HCII2005, 2005). In fact, several new PIDs have the capability to communicate with other IT devices without human intervention (HCII2205, 2005).

Now, we will explore some current applications of PIDs and some possible future applications to see where ergonomic and workstation issues might emerge. Millions and millions of cell phones are in use worldwide, and cell phones are a good representative of the PID. There is virtually nowhere in the world where cell phones cannot be used (with a few exceptions), and there is virtually no one (with adequate finances) who cannot access a cell phone and connect with the world. Cell phones can have many capabilities including telephoning, e-mailing, texting, auditory streaming, internet surfing, walkie-talkie communicating, photographing, video camera picturing and recording, and television/cable/satellite broadcast receiving. Even with all of these built in features, the size of cell phones is shrinking. As cell phones shrink, the manual, hand interfaces are getting smaller, as are the visual displays. With their small size cell phone can be carried easily and can be used when a person is walking, sitting, running, lying down, or hanging upside down. What are the ergonomic concerns with their use, and in particular the workstation design issues?

Small manual interfaces and displays make the accurate and comfortable application of perceptual-motor skills difficult (Albers & Kim, 2002; Hinckley, 2003; Haggerty & Tarasewich, 2005; Myers & Wobbrock, 2005). A cell phone can be held in one hand (this hand becomes the workstation) and then be manipulated with the other hand. In some instances, the hand holding the cell phone is also used to manipulate the manual interface. Either of these situations leads to workstation conditions where the users cannot apply their highest level of perceptual-motor skills. For example, they cannot use both hands for inputting into the interface, the device is held at an awkward angle for manual inputting, or the posture of the trunk is unstable which limits the capacity to use the hands effectively.

Due to this, a preferred form of input to control the action of the PID is speech (Gamm & Haeb-Umback, 1995; Karat, Vergo, & Nabamoo, 2003), while the displays are typically a combination of visual and auditory information. These interfaces may lead to ergonomic problems of overuse of the voice, increased duration of mental concentration, increased eyestrain, visual discomfort, and increased error rates of the communication with the devices. No body of research data tells us if these problems are or will become prevalent among PID users or if there will be long-term comfort, health, and performance effects. For now, we can only conjecture about the possibilities for problems. It is clear that many people are using PIDs and other computer-based technologies many hours per day, and this increased extent of use will likely lead to discomfort and some health effects for the voice, ears, upper extremities, and mental stress (Eklof, Gustafsson, Hagberg, & Thomee, 2005).

Now we will discuss workstation issues with PIDs, using the cell phone as an example. We will go back to the airport and I will use my cell phone to communicate with my office. At this time, I am standing in a long line at the airline check in counter. My cell phone has voice activation and control capability. I ask my cell phone for my e-mail service provider, and my e-mails come up on the display screen. I am using my thumb to scroll

through the e-mails one at a time. I hold the cell phone close to my face to enhance my ability to read the display screen. I shuffle forward as the line moves toward the airline counter (automated check in station). I am standing, and I have no postural support for my back, buttocks, and legs. It is true that I would be standing in line feeling the postural strain even if I were not using my cell phone. However, my cell phone use adds extra postural loading, since I am manually (vocally) manipulating the interface to operate the e-mail processing. The small visual display may create eyestrain, and if I talk on the cell phone for too long, it may create voice strain.

One way to reduce the load would be if I had a workstation where I could sit and support my back, buttocks, legs, and even my arms and elbows as I manually manipulate the interface. I could use a vocal and auditory interface with a head mounted earplug and microphone (headset), and thus eliminate the manual input and visual display viewing. The headset is a workstation improvement that reduces the manual manipulation and some of the postural loading on the upper extremities and back. The headset adds some weight to the head, and this increases loading on the neck, shoulders, and back. One difficulty with using an auditory interface is the high ambient noise in the airport lounge that interferes with and masks the cell phone's auditory signals. To reduce this auditory interference, a helmet with acoustical privacy could serve as a workstation improvement. However, a helmet adds substantial weight to the head, which puts increased loading on the neck, shoulders, and back. A major workstation improvement for users waiting in line, that would reduce the postural loading when using a cell phone, is to provide chairs so a user can sit down. A chair provides postural support for the back, buttocks, and legs. A chair on casters or wheels that can be scooted along as the line moves forward could be beneficial. However, this could increase the congestion in the lounge area as people are moving through the line. This example illustrates the new loads that may be added to technology users' lives by the use of PIDs. We will reiterate some of the advice from the laptop section as this is also applicable to the use of PIDs and other portable devices.

To reduce the undesirable effects of a lack of a workstation that can lead to the users' discomfort the following recommendations are given:

1. If you are using a laptop on your lap, find a work area where you can put the laptop on a table (rather than on your lap). Then arrange the work area as closely as possible with the recommendations presented above in the IEA laptop guidelines (Saito, 2000).
2. Never work in poor postural conditions for more than 30 minutes continuously. Take at least a 5-minute break (preferably 10 minutes) away from the laptop/PID use, put the device down (away), get up and stretch for 1 minute or more, and then walk for 2–3 minutes (unless you are already walking in which case you should sit down for 5 minutes). If you are using a handheld PID in a standing position, then during your break put it away, do 1 minute of stretching, and then sit down for 4 minutes. That may mean sitting on the floor, but preferably, you will sit where you can support your back (against a wall, or a seat back).
3. Buy equipment that provides the best possible input interfaces and displays (screens, headphones, typing pads). Since these devices are small, the perceptual motor requirements for their use are much more difficult (sensory requirements, motion patterns, skill requirements, postural demands, and force demands). Therefore, screens should provide easily readable characters (large, understandable), and input buttons should be easy to operate (large, properly spaced, easily accessible, low force).
4. Do not use these devices continuously for more than 30 minutes.

THE CHAIR AS A CRITICAL ELEMENT OF THE WORKSTATION

It was not until the last 40 years that sitting posture and chairs (seats) became topics for scientific research, especially for ergonomics and orthopedics. Studies have revealed that the sitting position, as compared to the standing position, reduces static muscular efforts in legs and hips, but increases the physical load on the intervertebral discs in the lumbar region of the spine.

The debate over what constitutes proper-seated posture is not yet fully resolved. Is an upright-seated posture most healthy, or is a relaxed posture with a backward-leaning trunk healthier? Interesting experiments by the Swedish surgeons Nachemson and Elfstrom (1970) and Andersson and Ortengreen (1974) offered some guidance about this. These authors measured the pressure inside the intervertebral discs as well as the electrical activity of the back muscles in relation to different sitting postures. When the backrest angle of the seat was increased from 90 to 120 degrees, subjects exhibited an important decrease of the intervertebral disc pressure and of the electromyographic activity of the back. Since heightened pressure inside intervertebral discs means that they have more stress, it was concluded that a sitting posture with reduced disc pressure is more healthy and desirable.

The results of the Swedish studies indicated that leaning the back against an inclined backrest transfers some of the weight of the upper part of the body to the backrest. This reduces considerably the physical load on the intervertebral discs and the static strain of the back and shoulder muscles. Thus, some computer users seem to instinctively get into the proper posture when they leaning backwards.

Most ergonomic standards for computer workstations are based on a more traditional view about a healthy sitting posture. Mandal (1982) reported that the "correct seated position" goes back to 1884 when the German surgeon Staffel recommended the well-known upright position. Mandal stated:

> But no normal person has ever been able to sit in this peculiar position (upright trunk, inward curve of the spine in the lumbar region and thighs in a right angle to the trunk) for more than 1–2 minutes, and one can hardly do any work as the axis of vision is horizontal. Staffel never gave any real explanation why this particular posture should be better than any other posture. Nevertheless, this posture has been accepted ever since quite uncritically by all experts all over the world as the only correct one.

The sitting posture of students in the lecture hall or of any other audience is very seldom a "correct upright position of the trunk." On the contrary, most people lean backward (even with unsuitable chairs) or, in some cases, lean forward with elbows resting on the desk. These two preferred trunk positions are probably associated with a substantial decrease of intervertebral disc pressure, as well as lessened tension of muscles and other tissues in the lumbar and thoracic spine (Nachemson & Elfstrom, 1970; Andersson & Ortengreen, 1974). Thus, sitting in general and particularly when using a computer is probably most comfortable and healthy when leaning back, or when the arms are supported while leaning forward.

Poorly designed chairs can contribute to computer user discomfort. Chair adjustability in terms of height, seat angle, and lumbar support helps to provide trunk, shoulder, neck, and leg postures that reduce strain on the muscles, tendons, and discs. The postural support and action of the chair help maintain proper-seated posture and encourage good movement patterns. A chair that provides swivel action encourages movement while backward tilting increases the number of postures that can be assumed.

The chair height should be adjustable so that the computer operator's feet can rest firmly on the floor with minimal pressure beneath the thighs. The minimum range of adjustment for seat pan height should be between 38 and 56 cm (15–22 inches) to accommodate a wide range of stature (BSR/HFES, 2005).

To enable short users to sit with their feet on the floor without compressing their thighs, it may be necessary to add a footrest. A well-designed footrest has the following features:

1. It is inclined upwards slightly (about 5–15 degrees).
2. It has a nonskid surface.
3. It is heavy enough that it does not slide easily across the floor.
4. It is large enough for the feet to be firmly planted.
5. It accommodates persons of different stature.

The seat pan is where the person sits on the chair. This part of the chair directly supports the weight of the buttocks. The seat pan should be wide enough to permit operators to make slight shifts in posture from side to side. This not only helps to avoid static postures, but also accommodates a large range of individual buttock sizes. The seat pan should not be overly U-shaped because this can lead to static sitting postures. The minimum seat pan width should be 46 cm (18 inches), and the depth between 38 and 43 cm (15–17 inches) (BSR/HFES, 2005; ANSI/HFES-100, 1988). The front edge of the seat pan should be well rounded downward to reduce pressure on the underside of the thighs that can affect blood flow to the legs and feet. This feature is often referred to as a "waterfall" design. The seat needs to be padded to the proper firmness that ensures an even distribution of pressure on the thighs and buttocks. A properly padded seat should compress about one-half inch to one inch when a person sits on it.

Some experts feel that the seat front should be elevated slightly (up to seven degrees), while others feel it should be lowered slightly (about five degrees). There is some disagreement among the experts about the correct answer. Due to this disagreement, many chairs allow for both front and backward angling of the front edge of the seat pan. The operator can then angle the chair's front edge to a comfortable position. The BSR/HFES (2005) guidelines recommend a user adjustable range of at least six degrees of forward and backward adjustment with at reclined position of three degrees. The seat pan height and angle adjustments should be accessible and easy to use from a seated position.

The tension and tilt angle of the chair's backrest should be adjustable. Inclination of chair backrest is important for operators to be able to lean forward or back in a comfortable manner while maintaining a correct relationship between the seat pan angle and the backrest inclination. A backrest inclination of about 110 degrees is considered an appropriate posture by many experts. However, studies have shown that operators may incline backwards as much as 125 degrees, which also is an appropriate posture. Backrests that tilt to allow an inclination of up to 125 degrees are therefore a good idea. The backrest tilt adjustments should be accessible and easy to use. An advantage of having an independent tilt angle adjustment is that the backrest tilt will then have little or no effect on the front seat height or angle. This also allows operators to shift postures readily. BSR/HFES (2005) recommends backrest adjustment of up to 105 degrees must be achievable with up to 120 degrees preferred.

Chairs with high backrests are preferred since they provide support to both lower back and the upper back (shoulder). This allows employees to lean backward or forward, adopting a relaxed posture and resting the back and shoulder muscles. A full backrest with a height around 45–51 cm (18–20 inches) is recommended. BSR/HFES (2005) recommends a backrest be at least 45 cm (18 inches) high. To prevent back strain it is also recommended that chairs have lumbar (midback) support, since the lumbar region is one of the most highly strained parts of the spine when sitting. BSR/HFES (2005) recommends that the width of the backrest should be at least 36 cm (14 inches).

For most computer workstations, chairs with rolling castors or wheels are desirable: They are easy to move and facilitate postural adjustment, particularly when the operator has to reach for equipment or materials that are on the secondary working surfaces. Chairs should have five supporting legs.

Another important chair feature is armrests. Both pros and cons to the use of armrests at computer workstations have been advanced. On the one hand, some chair armrests can present problems of restricted arm movement, interference with the operation of input devices, pinching of fingers between the armrest and table, restriction of chair movement, such as under the worktable, irritation of the arm or elbows due to tissue compression when resting on the armrest, and adoption of awkward postures. Properly designed armrests can overcome the problems mentioned above. Armrests can provide support for resting the arms to prevent or reduce arm, shoulder, and neck fatigue. Removable armrests are an advantage because they provide greater flexibility for individual operator preference. For specific tasks such as using a numeric keypad, a full armrest can be beneficial in supporting the arms. Many chairs have height adjustable armrests that are helpful for operator comfort, and some allow for adjusting the angle of the armrests as well.

ADDITIONAL WORKSTATION CONSIDERATIONS

Providing the capability for the screen to swivel and tilt up and down gives the user the ability to better position the screen for easier viewing. Reorientation of the screen around its vertical and horizontal axes can help to position a screen to reduce screen reflections and glare. Reflections can be reduced by simply tilting the display slightly back or down, or to the left or right away from the source of glare. The perception of screen reflections depends not only upon screen tilt, but also upon the operator's line of sight.

An important component of the workstation that can help reduce musculoskeletal loading is a document holder. When properly designed, proportioned, and placed, document holders reduce awkward inclinations of the head and neck and frequent movements of the head up and down and back and forth. They permit source documents to be placed in a central location at the same viewing distance as the computer screen. This eliminates needless head and neck movements and reduces eyestrain. In practice, some flexibility about the location, adjustment, and position of the document holder should be maintained to accommodate both task requirements and operator preferences. Dainoff (1982) showed the effectiveness of an inline document holder. The document holder should have a matte finish so that it does not reflect light.

Privacy requirements include both visual and acoustical control of the workplace. Visual control prevents physical intrusions, contributes to confidential/private conversations, and prevents the individual from feeling constantly watched. Acoustical control prevents distracting and unwanted noise (from machine or conversation) and permits speech privacy. While certain acoustical methods and materials, such as freestanding panels, are used to control general office noise level, they can also be used for privacy. Planning for privacy should not be made at the expense of visual interest or spatial clarity. For instance, providing wide visual views can prevent the individual from feeling isolated. Thus, a balance between privacy and openness enhances user comfort, work effectiveness, and office communications. Involving the employee in decisions of privacy can help in deciding the compromises between privacy and openness.

The use of a wrist rest when keying can help to minimize extension (backward bending) of the hand/wrist, but the use of a wrist rest for operator comfort and health has generated some debate because there are trade-offs for comfort and health. When the hand or wrist is resting on the wrist rest, there is compression of the tissue, which may create increased carpal canal pressure or local tissue ischemia. On the other hand, the wrist rest allows the hands and shoulders to be supported with less muscular tension, which is beneficial to computer operator comfort. At this time, there is no scientific evidence that the use of a wrist rest either causes or prevents serious musculoskeletal disorders of the hands, wrists, or shoulders. Thus, the choice to use a wrist rest should be based on employee comfort and performance considerations until scientific evidence suggests otherwise.

If used, the wrist rest should have a broad surface (five cm minimum) with a rounded front edge to prevent cutting pressure on the wrist and hand. Padding further minimizes skin compression and irritation. Height adjustability is important so that the wrist rest can be set to a preferred level in concert with the keyboard height and slope.

Arm holders are also available to provide support for the hands, wrists, and arms while keyboarding and have shown to be useful for shoulder comfort. The placement of the arm holder should not induce awkward postures in its use. The device should be placed within easy reach of the operator, especially when it will be used frequently during work.

When keyboard trays are used, they should allow for the placement of other input devices directly on the tray instead of on other working surfaces.

THE VISUAL ENVIRONMENT

Visual displays tend to be of two types, the cathode ray tube (CRT) and the diode matrix (LED/LCD) flat panels. Fixed workstations with personal computers tend to use CRTs because they are cheap, while laptop computers and hand-held devices are almost exclusively flat panel due to size and weight limitations. Both types of displays have characteristics that lead to problems from environmental influences. For instance, luminance sources in the environment that fall on the screen wash out characters on the screen, and the accumulation of dust particles on the screen may distort images. These conditions affect the ability to read the screen and can lead to visual fatigue and dysfunction. Specific characteristics of the environment, such as illumination and glare, have been related to computer-operator vision strain problems. The main visual functions involved in computer work are accommodation, convergence, and adaptation.

The alignment of lighting in relation to the computer workstation, as well as levels of illumination in the area surrounding a computer workstation, have been shown to influence the ability of the computer operator to read hardcopy and the computer screen (Cakir et al., 1979; Stammerjohn et al., 1981; Dainoff, 1983; Grandjean, 1987). Readability is also affected by the differences in luminance contrast in the work area. The level of illumination affects the extent of reflections from working surfaces and from the screen surface. Mismatches in these characteristics and the nature of the job tasks have been postulated to cause the visual system to overwork and lead to visual fatigue and discomfort (Cakir et al., 1979; NAS, 1983).

Generally, it has been shown that excessive illumination leads to increased screen and environmental glare, and poorer luminance contrast (Gunnarsson & Ostberg, 1977; Läubli, Hünting, & Grandjean, 1981; Ghiringhelli, 1980). Several studies have shown that screen and working surface glare are problematic for visual disturbances (Gunnarsson & Ostberg, 1977; Läubli et al., 1981; Cakir et al., 1979; Stammerjohn et al., 1981). Research by van der Heiden, Braeuninger, and Grandjean (1984) showed that computer users spend a considerable amount of their viewing time looking at objects other than the screen. Bright luminance sources in the environment can produce reflections and excessive luminance contrasts that may create excessive pupillary response, which leads to visual fatigue.

All surfaces within the visual field of an operator should be of a similar order of brightness as much as practical to achieve. The

temporal uniformity of the surface luminance is as important as the static spatial uniformity. Rhythmically fluctuating surface luminance in the visual field is distracting and reduces visual performance. Such unfavorable conditions prevail if the work requires the operator to glance alternately at a bright and then a dark surface, or if the light source generates an oscillating light. Sometimes fluorescent lights can have a noticeable stroboscopic effect on moving reflective objects, and sometimes when reflected from a computer CRT screen. When fluorescent tubes wear out or are defective, they develop a slow, easily perceptible flicker, especially at the visual periphery. Flickering light is extremely annoying and causes visual discomfort.

Lighting is an important aspect of the visual environment that influences computer screen and hardcopy readability, glare on the screen, and viewing in the general environment.

The intensity of illumination, or the illuminance being measured, is the amount of light falling on a surface. In practice, this level depends on both the direction of flow of the light and the spatial position of the surface being illuminated in relation to the light flow. Illuminance is measured in both the horizontal and vertical planes. At computer workplaces, both the horizontal and vertical illuminances are important. A document lying on a desk is illuminated by the horizontal illuminance, whereas the computer screen is illuminated by the vertical illuminance. In an office that is illuminated from overhead luminaries, the ratio between the horizontal and vertical illuminances is usually between 0.3 and 0.5. If the illuminance in a room is said to be 500 lux, this implies that the horizontal illuminance is 500 lux while the vertical illuminance is between 150 and 250 lux (0.3 to 0.5 of the horizontal illuminance).

The illumination required for a particular task is determined by the visual requirements of the task and the visual ability of the employees concerned. The illuminance in workplaces that use computer screens should not be as high as in workplaces that exclusively use hardcopy. Lower levels of illumination will provide better computer-screen image quality and reduced screen glare. Illuminance in the range of 300 to 700 lux measured on the horizontal working surface (not the computer screen) is normally preferable. The lighting level should be set up according to the visual demands of the tasks performed. For instance, higher illumination levels are necessary to read hardcopy and lower illumination levels are better for work that just uses the computer screen. Thus, a job in which both a hardcopy and a computer screen are used should have a general work-area illumination level of about 500 to 700 lux; a job that requires reading only the computer screen would have a general work area illumination of 300 to 500 lux.

Conflicts can arise when both hardcopy and computer screens are used by different employees who having differing job task requirements or differing visual capabilities and are working in the same room. As a compromise, room lighting can be set at the lower level (300 lux) or intermediate level (500 lux) and additional task lighting for hardcopy tasks can be provided at each workstation as needed. Such additional lighting must be carefully shielded and properly placed to avoid glare and reflections on the computer screens and adjacent working surfaces of other employees. Furthermore, task lighting should not be too bright in comparison to the general work area lighting, since the contrast between these two different light levels may produce eyestrain.

The surface of the computer screen reflects light and images. The luminance of the reflections decreases character contrast and disturbs legibility; it can be so strong that it produces a glare. Image reflections are annoying, especially since they also interfere with focusing mechanisms; the eye is induced into focusing between the text and the reflected image. Thus, reflections are also a source of distraction. Stammerjohn et al. (1981), as well as Elias and Cail (1983), observed that bright reflections on the screen are often the principal complaint of operators.

Luminance is a measure of the brightness of a surface, the amount of light leaving the surface of an object, either reflected by the surface (as from a wall or ceiling), emitted by the surface (as from the CRT characters), or transmitted (as light from the sun that passes through translucent curtains). High intensity luminance sources (such as windows) in the peripheral field of view should be avoided. In addition, a balance among luminance levels within the computer user's field of view should be maintained. To reduce environmental glare, the luminance ratio within the user's near field of vision should be approximately 1:3 and within the far field of vision should be approximately 1:10. For luminance on the screen itself, the character-to-screen background-luminance contrast ratio should be at least 7:1. To give the best readability for each operator, it is important to provide screens with adjustments for character contrast and brightness. These adjustments should have controls that are obvious and easily accessible from the normal working position (e.g., located at the front of the screen).

Experts have traditionally recommended a viewing distance between the screen and the operator's eye of 45–50 cm, but no more than 70 cm. However, experience in field studies has shown that users may adopt a viewing distance greater than 70 cm and still be able to work efficiently and comfortably. Thus, viewing distance should be determined in context with other considerations. It will vary depending upon the task requirements, computer screen characteristics, and individual visual capabilities. For instance, with poor screen or hardcopy quality, it may be necessary to reduce viewing distance for easier character recognition. Typically, the viewing distance should be 50 cm or less because of the small size of the characters on the computer or IT device screen.

THE AUDITORY ENVIRONMENT

As the use of PIDs in a wide variety of environments increases, the need for a quiet auditory environment that allows the user to easily hear and speak at a normal loudness level becomes more important. Crowded, noisy environments detract from users' abilities to hear. Many PIDs have auditory interfaces, such as headphones, which can concentrate the primary auditory signal and block out some of the environmental noise. One concern is that users may increase the loudness beyond levels that are safe for the auditory sensory system (CDC, 2003). High ambient noise in the environment leads to the need for greater mental concentration (attention) to the auditory signals, in-

creased intensity of the primary auditory signal (up to levels that may cause temporary auditory threshold shifts), and mood disturbances (irritation, anger, and discouragement; Crocker, 1997). These effects can result in increased mental fatigue, psychosocial stress, and long-term exposures may lead to discomfort and health consequences.

While headphones may provide benefits to concentrate the primary auditory signal, they also have the drawback of providing too much auditory energy to the ears. Prolonged exposure to load primary auditory signals can cause the adverse effects described above. To protect the auditory sensory system when using PIDs, we proposed the following:

1. Find environments to use your PID where the ambient auditory levels are 50 dba or less.
2. Do not use your PID for more than 30 minutes without taking a break where you can rest your eyes and ears. You should rest for at least 10 minutes before you start using your PID again. As your total use of the PID increases over the course of a day, the rest breaks should become longer. While the literature suggests that taking a sufficient break after 30 minutes of PID use should be sufficient for your eyes to recover (given a proper visual environment as defined above), the literature does not provide guidance on the maximum amount of time of PID use before a break is necessary to provide recovery for your ears and mental mood. Nor does the literature indicate the minimum amount of break time necessary to achieve auditory or mental recovery.
3. Keep the volume of the headphones of the PID at the lowest level necessary to hear the primary auditory signal properly. In no instance exceed 85 dba, and do not exceed 30 minutes of continuous exposure (see 2 above). It is best to not exceed 70 dba, and if you adhere to 1 above, you will not have to exceed 60–70 dba if you have normal hearing.
4. Be courteous when using your microphone or cell phone. Do not talk directly at other people such that your conversation may produce masking noise that interferes with their conversations. Keep your voice as low as possible that allows your interface or listener to understand your signal.

ERGONOMIC IMPLEMENTATION ISSUES

Implementing a workplace change such as improving workstation design or work methods is a complex process because it impacts many elements of the work system (Derjani & Smith, 1999; Smith & Carayon, 1995; Hagberg et al., 1995; Smith & Sainfort, 1989; Carayon & Smith, 2000; Smith & Derjani-Bayeh, 2003). Managers, designers, and engineers often like to believe that technological enhancements are easy to make, and that performance and health improvements will be immediate and substantial. Proper implementation involves changes in more than the workstation; for instance, there needs to be consideration of the work organization, job content, task improvements, job demands, and socialization issues. Planning for change can help the success of implementation and reduce the stress generated by the change. The success of implementing change depends heavily on the involvement and commitment of the concerned parties, in particular management, technical staff, support staff, first line supervision, and the employees.

There is universal agreement among change management experts that the most successful strategies for workplace improvements involve all elements (subsystems) of the work system that will be affected by the change (Hendrick, 1986; Lawler, 1986; Smith & Carayon, 1995; Carayon & Smith, 2000). Involvement assumes that there is an active role in the change process, not just providing strategic information. Active participation generates greater motivation and better acceptance of solutions than passively providing information and taking orders. Active participation is achieved by soliciting opinions and sharing authority to make decisions about solutions. However, one drawback of active participation is the need to develop consensus among participants who have differing opinions and motives. This usually takes more time than traditional decision making, and can bring about conflict among subsystems. Another drawback is that line employees often do not have the technical expertise necessary to form effective solutions.

Participative ergonomics can take various forms, such as design decision groups, quality circles and worker-management committees. Some of the common characteristics of these various programs are employee involvement in developing and implementing ergonomic solutions, dissemination and exchange of information, pushing ergonomics expertise down to lower levels, co-operation between experts and nonexperts. One of the characteristics of participatory ergonomics is the dissemination of information (Noro, 1991). Participative ergonomics can be beneficial to reduce or prevent resistance to change because of the information provided to the various members of the organization concerned with the new technology. Uncertainty and lack of information are two major causes of resistance to change and have been linked to increased employee stress. If employees are informed about potential ergonomics changes in advance, they are less likely to actively resist the change.

Training computer users about how the new workstation functions and operations are important especially if the adjustment controls are neither obvious nor intuitive. Hagberg et al. (1995) indicated that employee training is a necessary component to any ergonomic program for reducing work-related musculoskeletal disorders. Green and Briggs (1989) found that adjustable workstations are not always effective without appropriate information about benefits of adjustments and training in how to use the equipment. Hagberg et al. (1995) suggested the following considerations for ergonomics training programs:

1. Have employees involved in the development and process of training. Using employee work experiences can be helpful in illustrating principles to be learned during training. In addition, using employees as instructors can be motivational for the instructors and learners.
2. Use active learning processes where learners participate in the process and apply hands-on methods of knowledge and skill acquisition. This approach to learning enhances acquisition of inputs and motivation to participate.
3. Apply technology, such as audio-visual equipment and computers, to illustrate principles. Much like active processes,

technology provides opportunities for learners to visualize the course materials and to test their knowledge dynamically and immediately.

4. Use of on-the-job training is preferred over classroom training. Either can be effective when used together.

SUMMARY OF RECOMMENDATIONS

A computer workstation (work area) is comprised of the computer, input and output interfaces, the furniture where the computer is used, and the physical environment in which the computer is used. The design of these elements and how they fit together play a crucial role in user performance and in minimizing potential adverse discomfort and health consequences. The recommendations presented in this chapter address the physical environment and implementation issues. Important consideration should also be given to organizational factors and task related factors as they affect and depend on the individual. Unique situational factors need to be considered when an ergonomic intervention is being implemented. Generalizing these recommendations to all situations in which computers or PIDs are used is a mistake. The approach presented in this chapter emphasizes the adaptation of the work area to a user's needs so that sensory, musculoskeletal, and mental loads are minimized.

Positioning of computers in the workplace is important in order to provide a more productive work environment. Computers workstations should be placed at right angles to the windows, and windows should not be behind or in front of the operator. This will reduce the possibility of reflections on the screen, which can otherwise reduce legibility. In addition, bright reflections coming from light sources can be reduced by placing these light sources on either side of or parallel to the line of vision of the operator.

Moreover, illumination should be adapted to the quality of the source documents and the task required. This is done by having a high enough illumination to enhance legibility, yet low enough to avoid excessive luminance contrasts. Recommended levels range between 300 and 700 lux.

The keyboard and other input devices should be movable on the work surface, but stable when in use. A wrist rest or support surface of 15 cm in depth is recommended to rest the wrist and forearms. Computer workstations should allow for adjustments that promote good postures. A computer workstation without adjustable keyboard (input device) height and without adjustable height and distance of the screen is not reasonable for continuous work of more than 30 minutes. The controls for adjusting the dimensions of a workstation and chair should be easy to use. Such adjustability is particularly important at workstations and chairs used by more than one employee. Furthermore, sufficient space for the user's thighs, knees, and legs should be provided to allow for comfort and to avoid unnatural or constrained postures.

With regard to chair design, a backward-leaning posture allows for relaxation of the back muscles and decreases the load on the intervertebral discs. Chair seat pan height should be easily adjustable and fit a wide range of statures. The chair should have a full backrest that can incline backward of up to 130 degrees. In addition, the backrest should have a lumbar support and a slightly concave form at the thoracic level.

PIDs pose unique problems and ergonomic sense indicates that users need to take actions that will minimize the loads on their musculoskeletal system, sensory systems, and mental processes. PIDs have inherent ergonomic problems due to their ubiquitous applications, small size, and potential for users' over exposures to prolonged use. Using PIDs in the proper environment, limiting the extent of use by taking frequent rest breaks, and having the best possible interfaces is good ergonomic sense.

These recommendations are not exhaustively inclusive of all aspects of computer (and PID) workstation (environment) design. The chapter described other important workstation criteria in various sections.

References

Albers, M., & Kim, L. (2002). Information design for small-screen interface: An overview of web design issues for personal digital assistants. *Technical Communication, 49*(1), 45–60.

Andersson, B. J. G., & Ortengreen, R. (1974). Lumbar disc pressure and myoeletric back muscle activity. *Scandinavian Journal of Rehabilitation Medicine, 3*, 115–121.

ANSI/HFES. (1988). *American national standard for human factors engineering of visual display terminal workstations* (ANSI/HFS Standard No. 100-1988). Santa Monica, CA: The Human Factors Society.

Bergqvist, U., Knave, B., Voss, M., & Wibom, R. (1992). A longitudinal study of VDT work and health. *International Journal of Human-Computer Interaction, 4*(2), 197–219.

Bergqvist, U. O. (1984). Video display terminals and health: A technical and medical appraisal of the state of the art. *Scandinavian Journal of Work, Environment and Health, 10*(Suppl. 2), 87.

Bernard, P. B. (1997). *Musculoskeletal disorders and workplace factors DHHS.* (NIOSH) Washington, DC: National Technical Information Service. (NTIS No. 97-141)

BSR/HFES. (2005). *Human factors engineering of computer workstations* (BSR/HFES-100), Santa Monica, CA: The Human Factors and Ergonomics Society.

Cakir, A., Hart, D. J., & Stewart, T. F. M. (1979). *The VDT manual*. Darmstadt: Inca-Fiej Research Association.

Carayon, P., Smith, M. J., & Haims, M. C. (1999). Work organization, job stress, and work-related musculoskeletal disorders. *Human Factors, 41*, 644–663.

Carayon, P., & Smith, M. J. (2000). Work organization and ergonomics. *Applied Ergonomics, 31*, 649–662.

Centers for Disease Control. (2003). Third national health and nutrition examination survey (NHANES lll). Atlanta: Author.

Cohen, W. J., James, C. A., Taveira, A. D., Karsh, B., Scholz, J., & Smith, M. J. (1995). Analysis and design recommendations for workstations: A case study in an insurance company. *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting, 1, San Diego*, 412–416.

Crocker, M. J. (1997). Noise. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (2nd ed., pp. 790–827). New York: John Wiley & Sons, Inc.

Canadian Standards Association. (1989). *A guideline on office ergonomics: A national standard of Canada* (Document No. CAN/CSA-Z412-m89). Rexdale, Ontario: Author.

Dainoff, M. J. (1982). Occupational stress factors in visual display terminal (VDT) operation: A review of empirical research. *Behaviour and Information Technology, 1*(2), 141–176.

Dainoff, M. J. (1983, December). Video display terminals: The relationship between ergonomic design, health complaints and operator performance. *Occupational Health Nursing, 12*, 29–33.

Derjani-Bayeh, A., & Smith, M. J. (1999). Effect of physical ergonomics on VDT workers' health: A longitudinal intervention field study in a service organization. *International Journal of Human Computer Interaction, 11*(2), 109–135.

Eklof, M., Gustafsson, E., Habberg, M., & Thomee, S. (2005). ICT use and the prevalence of upper extremity symptoms and psychological symptoms among young adults in a prospective cohort study. In *Proceedings of the 11th International Conference on Human-Computer Interaction: Vol. 1. Las Vegas, July 22–27, 2005* [Electronic publication]. Mahwah, NJ: Lawrence Erlbaum Associates.

Elias, R., & Cail, F. (1983). Constraints et astreints devant les terminaux a ecran cathodique. (1109). Securite, Paris. Institut National de Recherche et de

Gamm, S., & Haeb-Umback (1995). User interface design of voice controlled consumer electronics. *Philips Journal of Research, 49*(4).

Ghiringhelli, L. (1980). Collection of subjective opinions on use of VDUs. In E. Grandjean (Ed.), *Ergonomical and medical aspects of cathode ray tube displays*. Zurich, Switzerland: Federal Institute of Technology.

Grandjean, E. (1979). *Ergonomical and medical aspects of cathode ray tube displays*. Zurich, Switzerland: Federal Institute of Technology.

Grandjean, E. (1984). Postural problems at office machine work stations. In E. Grandjean (Ed.), *Ergonomics and health in modern offices* (pp. 445–455). London: Taylor & Francis, Ltd.

Grandjean, E. (1987). Design of VDT workstations. In G. Salvendy (Ed.), *Handbook of human factors* (pp. 1359–1397). New York: John Wiley & Sons.

Grandjean, E., Hünting, W., & Nishiyama, K. (1982). Preferred VDT workstation settings, body posture and physical impairments. *Journal of Human Ergology, 11*(1), 45–53.

Grandjean, E., Hünting, W., & Nishiyama, K. (1984). Preferred VDT workstation settings, body posture and physical impairments. *Applied Ergonomics, 15*(2), 99–104.

Grandjean, E., Hünting, W., & Pidermann, M. (1983). VDT workstation design: Preferred settings and their effects. *Human Factors, 25*(2), 161–175.

Grandjean, E., Nishiyama, K., Hünting, W., & Pidermann, M. (1982). A laboratory study on preferred and imposed settings of a VDT workstation. *Behaviour and Information Technology, 1*, 289–304.

Green, R. A., & Briggs, C. A. (1989). Effect of overuse injury and the importance of training on the use of adjustable workstations by keyboard operators. *Journal of Occupational Medicine, 31*, 557–562.

Gunnarsson, E., & Ostberg, O. (1977). *Physical and emotional job environment in a terminal-based data system* (1977:35): Department of Occupational Safety, Occupational Medical Division, Section for Physical Occupational Hygiene.

Hagberg, M., Silverstein, B., Wells, R., Smith, M. J., Hendrick, H., Carayon, P., et al. (1995). *Work related musculoskeletal disorders (WRMSDs): A reference book for prevention*. London: Taylor & Francis, Ltd.

Haggerty, B., & Tarasewich, P. (2005). A new stylus-based method for text entry on small devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction: Vol. 4. Las Vegas, July 22–27, 2005* [Electronic publication]. Mahwah, NJ: Lawrence Erlbaum Associates.

HCII2005. (2005). *Proceedings of the 11th International Conference on Human-Computer Interaction, Las Vegas, July 2–27, 2005* [Electronic publication]. Mahwah, NJ: Lawrence Erlbaum Associates.

Heiden, G. H. van der, Braeuninger, U., & Grandjean, E., (1984). Ergonomic studies on computer aided design. In E. Grandjean (Ed.), *Ergonomic and health aspects in modern offices.* London: Taylor & Francis, Ltd.

Hendrick, H. (1986). Macroergonomics: A conceptual model for integrating human factors with organizational design. In O. Brown & H. Hendrick (Eds.), *Human factors in organizational design and management* (pp. 467–477). Amsterdam: Elsevier Science Publishers.

Hinckley, K. (2003). Input technologies and techniques. In J. A. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 151–168). Mahwah, NJ: Lawrence Erlbaum Associates.

Hirose, M., & Hirota, K. (2005). PUI (Perceptual User Interface) In *Proceedings of the 11th International Conference on Human–Compter Interaction: Vol. 5, Las Vegas, July 22–27, 2005* [Electronic publication]. Mahwah, NJ: Lawrence Erlbaum Associates.

Hultgren, G., & Knave, B. (1973). Contrast blinding and reflection disturbances in the office environment with display terminals. *Arbete Och Halsa.*

Hünting, W., Läubli, T., & Grandjean, E. (1981). Postural and visual loads at VDT workplaces: I. Constrained postures. *Ergonomics, 24*(12), 917–931.

Karat, C.-M., Vergo, J., & D. Nahamoo. (2003). Conversational interface technologies. In J. A. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 169–186). Mahwah, NJ: Lawrence Erlbaum Associates.

Läubli, T., Hünting, W., & Grandjean, E. (1981). Postural and visual loads at VDT workplaces: II. Lighting conditions and visual impairments. *Ergonomics, 24*(12), 933–944.

Lawler, E. E. (1986). *High-involvement management*. San Francisco, CA: Jossey-Bass Publishers.

Mandal, A. C. (1982). The correct height of school furniture. *Human Factors, 24*(3), 257–269.

Myers, B. A., & Wobbrock, J. O. (2005). Text input to handheld devices for people with physical disabilities. In *Proceedings of the 11th International Conference on Human-Computer Interaction: Vol. 4. Las Vegas, July 22–27, 2005* [Electronic publication]. Mahwah, NJ: Lawrence Erlbaum Associates.

Nachemson, A., & Elfstrom, G. (1970). Intravital dynamic pressure measurements in lumbar discs. *Scandinavian Journal of Rehabilitation Medicine*, (Suppl. 1).

NAS. (1983). *Video terminals, work and vision*. Washington, DC: National Academy Press.

Noro, K. (1991). Concepts, methods and people. In K. Noro & A. Imada (Eds.), *Participatory ergonomics* (pp. 3–29). London: Taylor & Francis, Ltd.

Ostberg, O. (1975, November/December). Health problems for operators working with CRT displays. *International Journal of Occupational Health and Safety, 6,* 24–52.

Saito, S., Piccoli, B., Smith, M. J., Sotoyama, M., Sweitzer, G., Villanuela, M. B. G., & Yoshitake, R. (2000). Ergonomics guidelines for using notebook personal computers. *Industrial Health, 38*, 421–434.

Smith, M. J. (1984). Health issues in VDT work. In J. Bennet, D. Case, J. Sandlin, & M. J. Smith (Eds.), *Visual display terminals* (pp. 193–228). Englewood Cliffs, NJ: Prentice Hall.

Smith, M. J. (1987). Mental and physical strain at VDT workstations. *Behaviour and Information Technology, 6*(3), 243–255.

Smith, M. J. (1997). Psychosocial aspects of working with video display terminals (VDT's) and employee physical and mental health. *Ergonomics, 40*(10), 1002–1015.

Smith, M. J., & Carayon, P. C. (1995). New technology, automation and work organization: Stress problems and improved technology imple-

mentation strategies. *International Journal of Human Factors in Manufacturing, 5*, 99–116.

Smith, M. J., & Sainfort, P. C. (1989). A balance theory of job design for stress reduction. *International Journal of Industrial Ergonomics, 4*, 67–79.

Smith, M. J., & Cohen, W. J. (1997). Design of computer terminal workstations. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (2nd ed., pp. 1637–1688). New York: John Wiley & Sons, Inc.

Smith, M. J., & Derjani-Bayeh, A. (2003). Do ergonomic improvements increase computer workers' productivity?: An intervention study in a call center. *Ergonomics, 46*(1), 3–18.

Smith, M. J., Carayon, P., & Cohen, W. (2003). Design of computer workstations. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 384–395). Mahwah, NJ: Lawrence Erlbaum Associates.

Stammerjohn, L. W., Smith, M. J., & Cohen, B. G. F. (1981). Evaluation of workstation design factors in VDT operations. *Human Factors, 23*(4), 401–412.

Part

III

DESIGNING HUMAN-COMPUTER INTERACTIONS

·17·

VISUAL DESIGN PRINCIPLES FOR USABLE INTERFACES

Everything Is Designed: Why We Should Think Before Doing

Suzanne Watzman
Watzman Information Design

Margaret Re
UMBC

Making Things Easier to Use and Understand: Thinking about the User's Experience **331**
Defining Visual Design **331**
The Design Process **331**
The Role of the Designer **332**
The Process of Good Design—How Do We Get There from Here? **332**
An Information-Design Process Is an Informed Design Process **332**
Phase 1: The Audit 333
Audit Questions A 333
Audit Questions B 333
Phase 2: Design Development 333
Phase 3: Implementation and Monitoring 333
Visual Design Principles **334**
Universal Principles of Visual Communication and Organization **334**
Visual Design Tools and Techniques **334**
The Five Criteria for Good Design 335
Visual Design Principles at Work 335
Typography 335
How the Human Eye Sees, and then Reads **337**
Typeface Size and Selection **337**
Serif and sans serif 337
Families of type 33
Variations in Letterforms **339**
Variations in Stress 339
Variations in Thick and Thin 339
Variations in Serifs 339
Typographic Guidelines **339**
Combining Typefaces 339
Contrast in Weight (Boldness) 340
Output Device and Viewing Environment 340
Letter Spacing and Word Spacing 340
Line Spacing/Leading 340
Line Length/Column Width 340
Justified vs. Ragged Right (FlushLleft) 340
Highlighting with Type 340

Decorative Typefaces 341
Black on White vs. White on Black and Dark on Light Background vs. Light on a Dark Background 341
Positive and negative type 341
Design Principles: Page Design 341
Building the Design of a Page 341
Gray page or screen 341
Chunking 341
Queuing 341
Filtering 341
Mixing modes 341
Abstracting 343
Other Page Design Techniques 343
White space 343
The grid 343
Field of vision 343
Proximity 343
The illusion of depth 344
Charts, Diagrams, Graphics, and Icons 344
Tables, Charts, Diagrams 344
Icons and Visual Cues 344
Illustrations and Photographs 345
Guidelines 346
Visuals should reinforce the message 346
Create a consistent visual language 346
Consider both function and style 346
Focus on quality vs.quantity 346
Work with a professional 346
Build a library to create visual consistency, organizational identity, and a streamlined process 347
Reinforce shared meaning (common visual language) 347
Color 347
Basic Principles of Color 347
Additive primaries 347
Subtractive primaries 347
How to Use Color 347
Less is more . . . useful and understandable 347
Create a color logic; use color coding 347
Create a palette of compatible colors 348
Use complementary colors with extreme caution 348
Decisions regarding color in typography are critical 348
Consider the viewing medium 348
Context is everything 348
Contrast is critical when making color choices 348
Quantity affects perception 348
Use color as a redundant cue when possible 348
We live in a global world, so when in Rome 348
Creating a System: Graphic Standards 348
What does a system cover? 350
Developing the system 350
Development 350
Implementation 350
Designing the Experience 350
Effective and appropriate use of the medium 351
The element of time 351
Consistent and Appropriate Visual Language 351
Navigational Aids 351
Graphics/icons 351
Metaphor 351
Color 351
Legibility 352
Readability 352
Guidelines 352
Use the analogy of a poster as a guide to design 352
Design for the most difficult common denominator 352
Avoid overuse of saturated colors 352
Consider different users' levels of skill 352
Be aware of the fatigue factor 352
Other differences to consider 352
Use the "squint test" to check the design 352
Challenges and Opportunities 352
Creating Your Own Guidelines 352

Take a moment and visualize Las Vegas at night. What kind of image does this conjure up for you? Flashing lights from all directions, a hotel's lighting display designed to outdo its neighbor as well as that of the casino signage down the street. At first glance, everything is exciting, colorful, and beautiful. Now add a fireworks display to your picture. More color, more excitement. Where do you look first? There is so much going on; it is hard to see it all, but you don't want to miss a thing. Your head turns in all directions. You look there; then, out of the corner of your eye, you see something else. Look over there! Now the fireworks are at their peak, and the noise gets even louder. Any conversation with companions is impossible, yet it is also impossible to focus on any one thing for more than a split second. You are overwhelmed and overloaded. Everything is screaming for your

attention. Can you manage to pay attention? For how long? Do you begin to shake your head in despair, and give up? Do you wish you were somewhere else—NOW?

MAKING THINGS EASIER TO USE AND UNDERSTAND: THINKING ABOUT THE USER'S EXPERIENCE

The previous description is unfortunately an accurate analogy of many users' experiences as they attempt to learn, work, play, and relax. New products, new services, and new technology with which you are unfamiliar can create confusion. Users of these new products, services, and technology are customers, electricians, grandparents, clerks, pilots, and students—you and me. And for most of us, it's a jungle out there! Las Vegas at night with fireworks, or, monitors that are winking, blinking, distracting, disturbing, overwhelming—and, after a short time, visually deafening. Now add voices coming from boxes. . .! Although this may seem like an exaggeration, for many this situation is exactly their experience. User interface design focuses on designing flexible environments that have a positive impact a user's ability to experience and interact with a product whether that product is a mobile communication device, website, information kiosk, or appliance. It involves creating environments that include strong navigational devices that can be understood intuitively and used effortlessly. Designers have a responsibility to create user experiences that are simple and transparent. To do their job well, they must advocate on behalf of the user, ensuring that the interfaces they design are not just merely exercises in technology but that they truly assist and guide the user from task to task, enabling work to be done, and ultimately improving quality of life. When designers succeed, their products can be used effortlessly and are even pleasurable to use. Good design does not needlessly draw attention to itself. It just works. This is the role of good design.

DEFINING VISUAL DESIGN

The nautilus shell is an example of the synthesis between form and function found in nature (Fig. 17.1). Its form is the result of evolution, which is both transparent and beautiful. The nautilus shell is a perfect analogy for design and the design process because it creates valuable user experiences and usable interfaces.

The word *design* functions as both a noun and a verb. Many people use it to refer to the outward appearance or style of a product. However, design also refers to a process—that of intentionally establishing a plan or system by which a task can be accomplished or a goal reached. It includes tangible and intangible systems in which objects or processes are coherently organized to include the environments in which these objects or processes function. Design affects all people in every aspect of what they do. Good design performs for people. It is concerned with economics and the transmission of ideas. The challenge presented to a design team is to plan a prototype with a clear purpose that is easy to use, meets user needs, addresses commercial considerations, and can be mass-produced. Its visual form, whether two- or three-dimensional, digital or analog, logically explains its purpose and efficiently leads the user through its function. Design is not a series of subjective choices based on personal preference, at best a cosmetic afterthought considered if time and money are leftover. Good design is the tangible representation of product goals. An iterative and interactive process that requires active learning, design unifies a wide range of disciplines. Good design is a significant activity that reveals multiple solutions to each problem. Design equally values different ways of thinking. It allows people with a variety of skills and learning abilities to work cooperatively to bring insights and expertise to problems and opportunities in order to better develop new and innovative solutions. Problems can be analyzed using a multitude of viewpoints and methods. Writing, drawing, statistical analysis, graphing, discussion, interviewing, personal observation, model-making, and diagramming are all legitimate methods for examination as the physical, social, and cultural contexts of possible answers are considered (Davis, Hawley, McMullan, & Spilka, 1997).

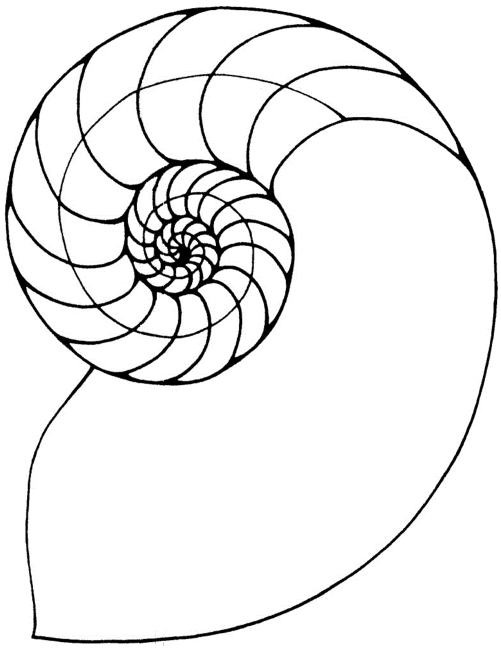


FIGURE 17.1. Nautilus Shell. The nautilus shell is an example of the synthesis between form and function found in nature. Its form is the result of evolution. The work of evolution is transparent as its beauty is admired. The nautilus shell is a perfect analogy for design and the design process to create valuable user experiences and usable interfaces.

THE DESIGN PROCESS

Design as a Catalyst for Learning, a publication funded in part by the National Endowment for the Arts, argued that effective design that responds to human problems uses the following steps (Fig. 17.2):

- *Problem identification and definition:* A need or problem is identified, researched, and defined.

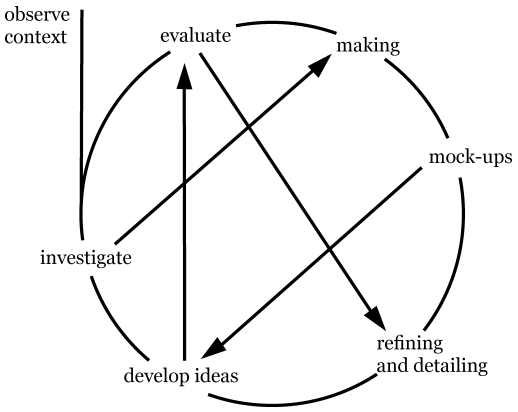


FIGURE 17.2. Interacting Design Loop. The interacting design loop developed by Richard Kimbell, founder of the Technology Education Research Unit at Goldsmiths College and presented in *Design as a Catalyst for Learning,* captures the divergent, iterative and cyclical nature of the design process.

- *Gathering and analyzing information:* The focus is on learning what is not known. Assumptions are questioned. Wide and broad research is used to locate information and generate ideas.
- *Determining performance criteria for a successful solution:* Research continues as imagery is selected. Rules are declared and what is known is specified.
- *Generating alternative solutions and building prototypes:* Multiple solutions are generated. A variety of methods for analysis, such as drawing, interviewing, modeling or evaluating statistics, are used.
- *Implementing choices:* Project content, scope, and intent are formally established. Initial possibilities are represented and presented as prototypes.
- *Evaluating outcomes:* Prototypes are assessed, tested, evaluated, and judged. The knowledge gained is incorporated into further studies and refinements.
- *Production:* A prototype, which is a synthesis of the initial solutions made using this process, and specifications are released for making multiples to a manufacturer.

THE ROLE OF THE DESIGNER

Visual design decisions are based on project goals, user perspective, and informed decision making. While many aspects of design are quantifiable, there are visual principles that are less measurable but equally important. Even though the necessary skills to become visually literate and make competent design decisions can be learned, design involves a highly specialized knowledge base. A unique combination of creativity and skill differentiates and makes one design more attractive and desirable than another. Both education and talent are necessary to apply the principles required to present information in its most accessible, useful, and pleasing form. The role of the designer in the development of interfaces for interactive products is to understand the product goals and ensure that information is approachable, useful, and desirable. In an environment in which the interface is the only tangible representation of a product and user perception determines product success, appropriate information presentation and visual design is key. Designers understand visual principles in context, and know how to apply them appropriately to create innovative, functional and aesthetically pleasing solutions.

THE PROCESS OF GOOD DESIGN— HOW DO WE GET THERE FROM HERE?

Interface designers are responsible for defining what the experience will be like when a product is used. While print media dictates that users encounter content in a largely predetermined sequence, an interface offers the user greater flexibility over how content can be accessed based on users' needs and wants. A successful interface can be easily navigated. Interface designers define, decide, and then create the experience for users, so that an experience with a product is useful, meaningful, even pleasant and empowering. The designer must maintain an attitude of unbiased discovery and empathy for the user. The designer must develop clearly defined goals in order to create a good design that includes an evaluation process that supports and enhances these goals, and includes the flexibility to respond to changes as the process continues and products evolve.

AN INFORMATION-DESIGN PROCESS IS AN INFORMED DESIGN PROCESS

An information-design process (IDP) is a method of visually structuring and organizing information to develop effective communication. Information design is not superficial or decorative, but is rather a merging of functional, performance-based requirements with the most appropriate form to present these requirements. A thoughtful, well-designed solution will,

- *Motivate users:* It psychologically entices an audience, convincing members that information and tasks can be successfully handled.
- *Increase ease of use and accessibility:* The effort needed to comprehend information is decreased. A clear path that aids in skimming and referencing text and gives easy access is provided.
- *Increase the accuracy and retention of the information:* Users learn and retain information better when it is visually mapped and structured in obvious and intuitive ways.
- *Focus on the needs of its users:* Multiple audiences have different requirements and styles of learning. Solutions should be developed that provide alternative means of accessing information for different types of users. An information-design approach is part of a process that incorporates research, design, testing, and training to produce useful, cost-effective solutions.

Phase 1: The Audit

The goal of the audit is to create a blueprint for the project, much like architectural drawings are developed before constructing a building (Fig. 17.3). The audit process begins by asking and answering a number of questions and acknowledging ongoing change and an ever-increasing palette of products and services. Questions are asked throughout the entire product life cycle, since the answers/design solutions reflect the user/use environment and affect the ongoing usefulness and value of the product. To create an eloquent design, continually ask and answer the following questions:

Audit Questions A

- *Who* are the product users?
- *How* will this product be used?
- *When* will this product be used?
- *Why* will this product be used?
- *Where* will this product be used?
- *How* will the process evolve to support this product as it evolves?

After the first set of questions are asked and answered, a second set of questions must be asked and answered:

Audit Questions B

- What is the most efficient, effective way for a user to accomplish a set of tasks and move on to the next set of tasks?
- How can the information required for product ease of use be presented most efficiently and effectively?
- How can the design of this product be done to support ease of use and transition from task to task as a seamless, transparent, and even pleasurable experience?
- What are the technical and organizational limits and constraints?

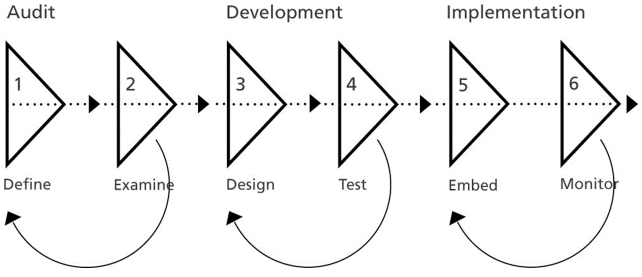


FIGURE 17.3. Information Design Process (IDP). IDP is phased to insure user and organizational needs are met. It is ongoing and iterative, throughout the lifecycle of a product. Any change can trigger a recycling of this process, to insure solutions remain appropriate and useful.

An audit focuses on discovery. Many disciplines and organizational resources must be consulted. Change is a given, since designers begin with assumptions and don't know all they need to know yet. The answers to their questions and their analysis in the context of organizational objectives provide the basis for the audit report, which serves as the guide in design development. The audit report can be as simple as a two-page list or as complex as a comprehensive hundred-page report. Since the goal is discovery, it includes every aspect of the organization concerned with the product-development cycle: project management, usability engineering, technical development, user support and documentation, visual communication and design, and content management. With these goals, the result is unbiased, accurate, comprehensive information that serves as the basis for design.

Phase 2: Design Development

The design-development phase uses the audit report as a guideline. This is an ongoing, iterative process with each iteration incorporating user test results to make the product appropriate to the particular set of needs. In reality, the length of this process is often defined and limited by real-world deadlines such as product release dates. The design-development phase includes design and testing. The designer or design team creates a number of solutions based on results and objectives determined by the audit report as well as other project specifics. Initially, design ideas should be very broad, incorporating many ideas and options no matter how unrealistic or unusual. As ideas are tested, user feedback incorporated, and other parameters defined, solutions naturally become more defined. Surviving design ideas are then based on solid information derived from user feedback, providing a strong basis for final design decisions. In the beginning, the focus is on high-level concepts and navigation. How will the product work? What will it feel like to use? As initial concepts are refined, design details become more specific. When the conceptual model and organizational framework are approved, the design of the look or product package begins. By the end of this phase, a prototype design to be carried out in implementation and monitoring is tested, approved, and specified.

Phase 3: Implementation and Monitoring

The implementation phase focuses on delivering what has been defined, designed, and documented in the preceding phases. It is the final part of a *holistic* process that defines everything necessary to make a product succeed on an ongoing basis. This includes not only the implementation of the design within the technology, but also any additional support such as the creation of training materials and other reinforcements that enhance use and productivity. Continuous monitoring is key to sustained success, because a successful product responds to evolving technology and user needs. This last phase is mostly consultative and ongoing throughout the product lifecycle in order to ensure that changes such as new technology and product developments are reflected in the product itself. These may in fact trigger another audit/design/testing cycle, although usually less extensive than the initial process. Though the implementation

phase is called "the last phase," it reveals the evolutionary process of design and development. The goal of ongoing monitoring of solutions is to be aware of changes in user needs, technology, and competition that impact user acceptance and satisfaction. Changes here often result in the need to reevaluate and redesign to incorporate this new knowledge gained.

VISUAL DESIGN PRINCIPLES

Interaction design bridges many worlds: that of visual design, information presentation, and usability with aesthetics. Donis A. Donis (1973a), in *A Primer of Visual Literacy,* argued that art and its meaning have dramatically changed in contemporary times from one that involved a concern with function to one that views the process of creating art as that of making emotional maps that spring from the province of the intuitive and subjective. This argument extends to design. To someone unskilled in creating effective communications, visual design is often understood as personal preference limited to style or appearance. However, any form of effective design is a result of rigorous study, a concern for organization and usability combined with knowledge of the basic design principles of harmony, balance, and simplicity. Visual design is in fact a form of literacy.

UNIVERSAL PRINCIPLES OF VISUAL COMMUNICATION AND ORGANIZATION

The principles of harmony, balance, and simplicity are related yet distinct in meaning and application. *Harmony* is the grouping of related parts, so that all elements combine logically to make a unified whole. In interface design, as with other categories of design, this is achieved when all design elements work in unity. Transitions from place to place are effortless and the techniques used to achieve this harmony are unnoticed by the user. Visual harmony achieves the same goal as musical harmony in which notes combine to create a chord. The golden section, also known as the "golden mean" or "golden rectangle," is one of the most widely used methods for creating harmony. Architects, artists, musicians, mathematicians, and designers have used the golden section extensively for centuries to create proportional relationships (Fig. 17.4).

Balance offers equilibrium or rest. Donis stated that equilibrium is the strongest visual reference (Donis, 1973b). It provides the equivalent of a center of gravity that grounds the page. Without balance, the page collapses, all elements are seen as dispersed, and content is lost. Balance requires continual modification from page to page because while each page is part of a greater system, elements can vary and all have visual weight. In the same way that a clown balancing on a ball while juggling objects of different weights must continually make adjustments for actions that are occurring, visual balance requires the same concerns and adjustments as in the physical world. Regardless of how a design is organized, it must achieve stability and unity in order for a user to feel comfortable with the solution. Balance can be

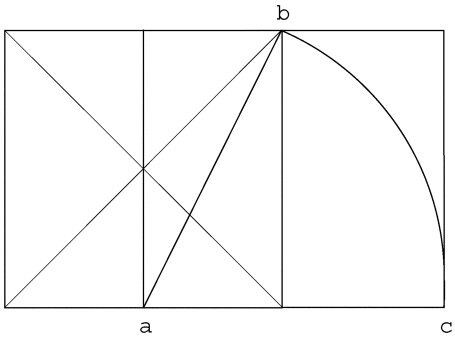


FIGURE 17.4. The Golden Rectangle. Divide a square to find the center point (a) from which length (ab) is found. From point (a) the length (ab) is swung as an arc to point (c) to create a rectangle that uses the proportions of the golden section.

achieved a number of ways. One obvious method uses *symmetry,* such as found on a page with text and image aligned on a centered axis. Deceptively simple, symmetry form is often considered easy to make; however, unless handled carefully a symmetrical composition can be predictable, boring, and static. *Asymmetry* employs nonaxial balance and uses contrast between elements such as weight, form, and color to create visual tension and drama. Both are valid approaches and require skill and knowledge of complex visual interaction to achieve.

Simplicity is the embodiment of clarity, elegance, and economy. A solution that offers simplicity is unambiguous and easily understood. It offers clarity working effortlessly devoid of unnecessary decoration. It appears deceivingly easy, accessible, and approachable, even though it may be conceptually rich. Simplicity involves distillation—every element is indispensable, if an element is removed, the composition falls apart. Achieving simplicity is no easy task. Two guidelines for creating simple design solutions are (a) "Less is more!" (attributed to Mies van der Rohe) and (b) "When in doubt, leave it out!" (Anonymous). The most refined design is direct and includes only essential elements. Removing any element breaks the composition rendering it unintelligible or radically different.

VISUAL DESIGN TOOLS AND TECHNIQUES

New technologies are rapidly being created that extend past the simple automation of tasks and communication; they are revolutionizing processes and the resulting products. Before the revolution brought about by electronic publishing technology, many disciplines such as writing, editing, design, publishing, and programming were discreet units that handled a defined step within a larger process. Today's new publishing environments encourage the possibility of a *renaissance* publisher—a person who can create, design, publish, and distribute. Yet the process used to arrive at successful solutions is very complex and extends past technical knowledge to include a mastery of visual and verbal language in order to build effective communi-

cation. Focus must be placed on the factors that determine success with constant evaluation and adjustment of these factors in light of new developments.

The Five Criteria for Good Design

Before any work begins, all participants in the process should have a clear understanding of the criteria for good design. The following questions are guidelines for evaluation of design solutions before, during, and after the process to ensure that all solutions remain valid as products, technology, and user needs evolve.

- ***Is it appropriate?*** Is the solution appropriate for the particular audience, environment, technology, and/or culture?
- ***Is it durable?*** Will the solution be useful over time? Can it be refined and transitioned as the product evolves and is redefined?
- ***Is it verifiable?*** Has the design been tested by typical users in the environment that it will be used in? Has feedback been properly evaluated and used to improve the product?
- ***Does it have impact?*** Does the design solution not only solve the problem, but also impact the look and feel, so that the user finds the product experience comfortable, useful, and desirable?
- ***Is it cost effective?*** Can the solution be implemented and maintained? Are individuals with the necessary skills and understanding to create, refine, and maintain the design available throughout the product's life? The cost of any design begins with the audit and design phases, but continues after the implementation phase to insure that it remains advantageous and cost-effective. The hard and soft costs of delivering the solution plus ongoing maintenance add up to the real design costs.

Visual Design Principles at Work

The following sections outline the disciplines and principles used to create quality design solutions. Each topic is worthy of extended study, because there is much to understand when evaluating how to effectively present information. As the design process evolves, insights, and information are discovered that impact a solution. It is optimistic to base solutions on an initial exercise because the very nature of process means discovering what is unknown yet critical. Therefore, all members involved in the design process must remain open and ready to incorporate new information, which may change or delay results, but more accurately reflect user needs.

For example, if a new feature is developed that changes a product's target audience from mid-level managerial to executive users, most methods for critical interaction and content delivery should be reconsidered. Executives have less time and need different information. The result might be a simpler interface with streamlined content that uses a larger typeface and a more conservative visual language. The most important principle to remember when thinking about design is that there are no rules, only guidelines. Everything is context sensitive. Always consider and respect the users.

Typography

In *The Elements of Typographic Style*, the poet and typographer, Robert Bringhurst (2005) described typography as frozen language. *Typography* is the visual representation of spoken and unspoken thought that allows an idea to be shared across time and distance independent of its creator. A functional and expressive art that shares many of the same concerns as writing and editing, typography involves organizing text so that its meaning is communicated according to an author's intent. In design, a literacy that provides an understanding of typography and how text can be structured in space is as important as a literacy that understands how to structure grammar in order to explicate content.

Typography is made from type, individual characters organized by visual characteristics into typefaces. Type is the smallest definable part of a design, much like a pixel is to a screen display. James Felici (2003), who has worked through evolutions in typesetting technologies in *The Complete Manual of Typography,* defined a font as the electronic file that contains the programming code needed to make the characters found in a typeface. Historically, a typeface consists of all the individual characters or glyphs at a given size: letterforms, punctuation, numbers, mathematical symbols, diacritical marks, and other accessory characters needed to fully compose a text. This definition serves as a reminder to read a text carefully and consider all needs before selecting a typeface and developing a presentation form (Fig. 17.5).

Effective typography is rational. It is concerned with clarity and comprehension; the ease in which characters and word shapes are recognized in reading environments and the ease in which they can be used. It extends past the shapes of individual letters and their potential combinations to include the relationships found between word and interword shapes, functional groupings that ultimately progress into issues of type weight, slope, width and scale, characteristics that act as interpretative devices in order to create influential and per-



FIGURE 17.5. Type Family. A type family is built around four core members: roman, bold, italic, and bold italic. Additional members may include typefaces whose weight and width are variants of the core group. A family can also contain expert sets that offer additional or alternate characters such as smallcaps, fractions and non-aligning numbers.

suasive form. If the principles of good typography can be understood and applied, then these same principles can be extended to more complex issues that follow such as page and product design. Typographic choice affects legibility and readability, the ability to easily see and understand what is on the page, in all media. Walter Tracy (1986), in *Letters of Credit: A View of Type Design,* offered the most useful definitions for legibility and readability. *Legibility,* the speed at which letters and the words built from them can be recognized, refers to perception. *Readability*, the facility and ease with which text can be read, refers to comprehension. Legibility and readability are related. Regardless of media, legibility and readability are determined by variables such as point size, letter pairing, word spacing, line length and leading, resolution, color, and organizational strategies such as text clustering. Together, legibility and readability comprise typography's functional aspects. Good typography, like good design, is invisible to the user—it just works (Fig. 17.6).

Selecting an appropriate typeface for a specific purpose and context requires experience and understanding (Fig. 17.7). With thousands of typefaces to choose from and numerous ways to manipulate them, finding the typeface best suited for an audience is not easy. With its lack of control, multiple media, and varied viewing contexts the current publishing environment makes this a complex task.

Typeface choice impacts whether and how a communication is read. Distinct typefaces and typographic styles create environments that influence a user's perception of text. The physical nature of the presentation itself helps determine content and acceptance. A typeface with extremely thick and thin strokes may appear sophisticated and readable in print but may look naïve and render text unreadable in a digital environment. Typefaces are frequently designed to solve issues of legibility and readability created by a technology. A typeface made for online use can increase page legibility, as well as the overall perception of approachability, quality of an interface, and ultimately product acceptance (Fig. 17.8, 17.9, & 17.10)

An informed selection can make reading enjoyable and effortless rather than frustrating and fatiguing. Though typography might seem to be an insignificant issue to a non-designer, it affects overall usability. A clear understanding of the concepts and principles that affect legibility and readability is crucial to determining effective typography.

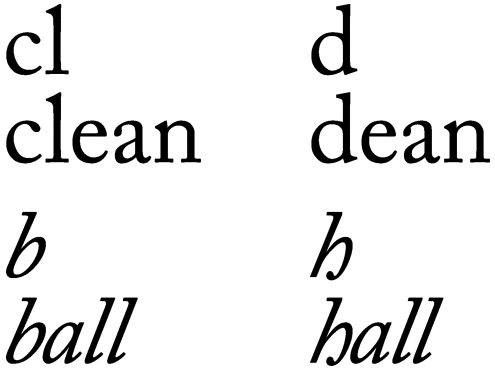


FIGURE 17.6. Legibility and Readability. The letters, letter pairs and words shown above are examples of what can happen if the designer is not sensitive to issues of legibility and readability.

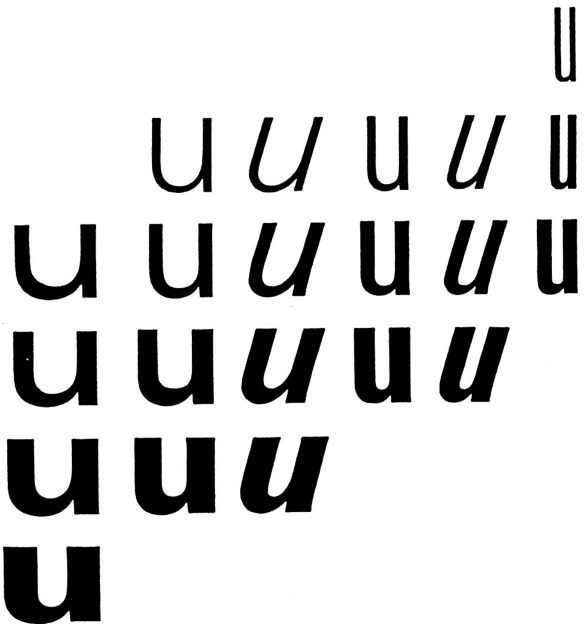


FIGURE 17.7. Univers "U". Univers, a type family designed by Adrian Frutiger and released for commercial use in 1954, is composed of twenty-one fonts that together offer a wide range of weights, widths and slopes that allows a text to be organized so that its form is visually coherent and easily read.

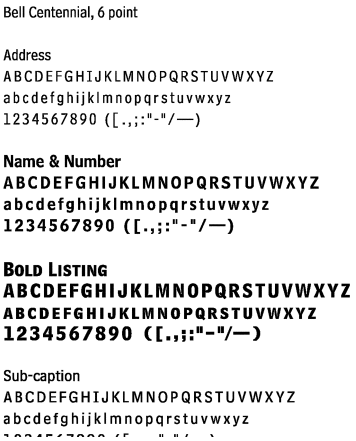


FIGURE 17.8. Bell Centennial: Technology specific typefaces. AT&T commissioned Bell Centennial, a typeface designed at a very small size for telephone directory use, in order to solve an industrial problem created by changing typesetting and printing technologies. The resulting type family designed for maximum legibility, readability and spatial efficiency provided the user with a clear information hierarchy. It reduced paper use and directory assistance calls. Here, Bell Centennial is shown at six point, the size at which it was intended to function.

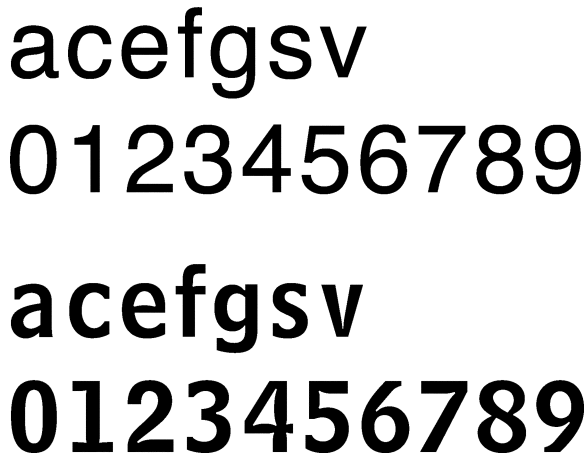


FIGURE 17.9. Bell Centennial: Technology specific typefaces. Bell Centennial's forms were opened to increase legibility and readability. Curved strokes were straightened and the horizontal and vertical juncture was notched so that it did they did not clog with ink when printed. Select letterforms shown at 32 point for Bell Centennial (bottom) and Helvetica (top) illustrate this.

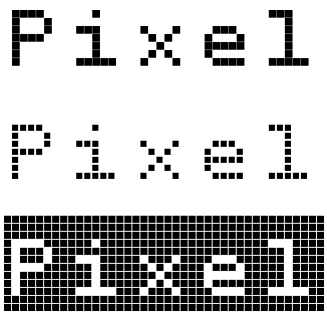


FIGURE 17.10. Pixel Font. Pixel fonts are screen fonts designed specifically for use with or as on screen navigational elements. Their block-like forms are the result of working with the actual pixels themselves.

HOW THE HUMAN EYE SEES, AND THEN READS

Herbert Spencer (1969) in *The Visible Word*, a publication with an objective of introducing and uniting those who research legibility with those who work with typography, presented that the eye uses both outline word shapes and their internal patterns to move along a text line and steps and jumps as it groups text to form comprehensible phrases of information. This motion of the eye during reading is known as "saccadic movement." Sometimes during this process, the eye regresses and returns to what has been read. Optimal typography allows for fewer backward movements. Spencer argues that possessing a mastery of reading mechanics is important to a mastery of content. Typeface selection directly affects this skill, making it easier or more difficult for the eye

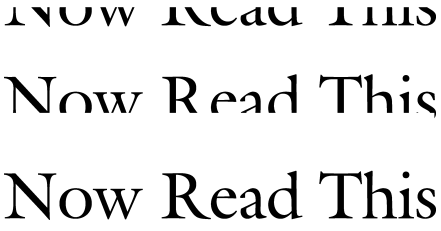


FIGURE 17.11. Now Read This. The phrase, Now Read This, is shown in full, cropped at the bottom, and from the top suggesting the importance of outline word shape and internal pattern.

to group, read, and understand information (Fig. 17.11, 17.12, & 17.13).

TYPEFACE SIZE AND SELECTION

Type size is given in points, a measuring system unique to typography. In digital typesetting systems, a point equals 1/72 of an inch. Type size cannot be determined by physically measuring a letterform because when type existed solely in metal, the technology in which it was first used, size was measured by the height of the metal block on which a letter sat. This is one of the reasons why the same letter repeatedly set in differing typefaces at the same point size appears dissimilar when compared. Lowercase letters set in the same point size with differing typefaces can appear larger or smaller in comparison to each other due to variations in x-height. Other variables such as stroke weight, letter width, and ascender and descender length influence size perception and help make some typefaces more or less readable and legible than others. Type size is also dependent on the resolution offered by output and viewing devices, color usage, context, and other design issues. It is crucial to understand not only the characteristics of a typeface but also usage context and application environment when selecting a typeface.

Serif and sans serif. Serif and sans serifs are general categories used for classifying type. *Serif* refers to a typeface with serifs, the short strokes that project off the end of letter strokes, as opposed to a *sans serif*, a typeface without serifs. It is debatable whether serif typefaces, conventionally used for setting text in print, are more readable than sans serif typefaces, which have an even stroke weight and more open counters that have proven useful for setting text for on-screen reading. And, while sans serif is considered easier to read on screen, serifs can be made equally legible if the appropriate typeface, size, and color is specified. Some designers think that, in print, serifs aid in character recognition and readability; they help differentiate individual letters creating horizontal lines for the eye to follow. This has not been proven conclusively. Other designers hold the view that "we read best what we read most." It's likely that this discussion will continue. Recent technological developments that subdivide a pixel into red, green, and blue elements on LCD screens have resulted in new technologies that create a better immersive environment

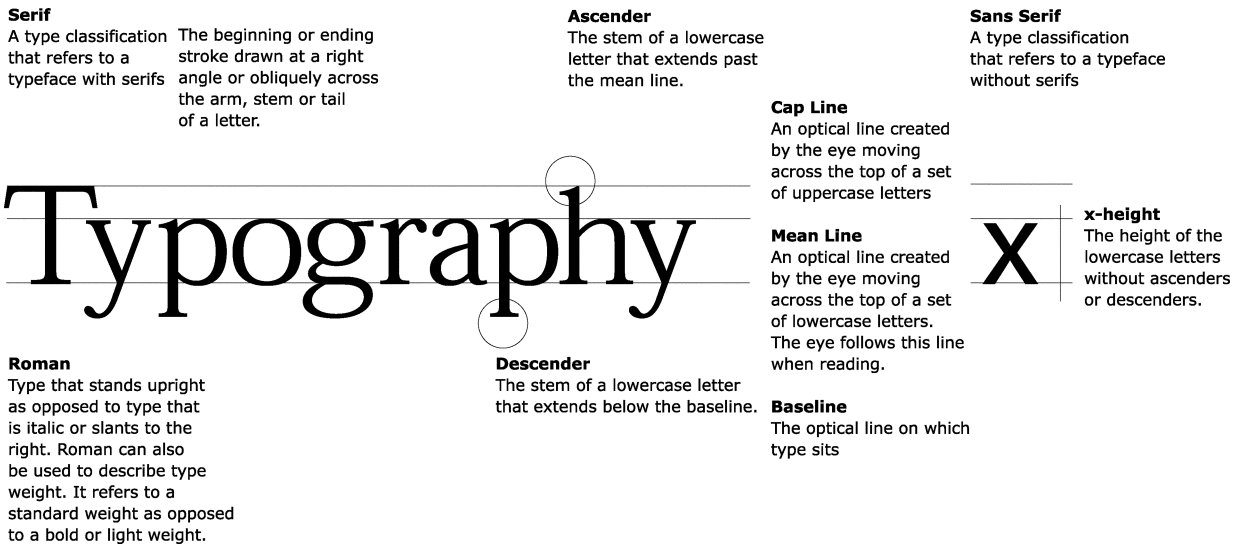


FIGURE 17.12. Anatomy of a letter. The typographic terms defined and illustrated above are used by designers in discussing principles that affect the legibility of type and overall quality of the communication.



FIGURE 17.13. Hqx Illustration. The x-height of a typeface (based on the actual height of a lower-case x) is a key characteristic when deciding the visual size of a typeface, particularly when readability is critical. While the above typefaces are the same point size, some seem larger (e.g., Helvetica) and easier to read than others (e.g., Serifa).

for online reading. This, in turn, will spark new explorations in typographic form and its presentation.

Families of type. Many design students first learn to classify typefaces into five chronological families or organizational groupings popularized by James Craig et al., (2006) in *Designing with Type*: (1) old style, (b) transitional, (c) modern, (d) Egyptian, and (e) contemporary. Classifying typefaces into these families makes it easier to understand the differences and similarities in structure and fitness. Like anything else, type design does not happen in isolation. These categories, as with those of many other classification systems, are formed around historical junctures in which the overall design of letterforms shifted dramatically in response to technological, political, cultural, aesthetic, and economic concerns. Typefaces within each set have passed through the tenures of metal, photocomposition, and been adapted for digital technologies. Many of the problems solved through these older forms have proven inspirational to contemporary type designers who as their predecessors did, continue to explore new forms for new purposes.

Verdana a sans serif screen font designed by Matthew Carter, whose roots lie in the Industrial Revolution, was commissioned by Microsoft and released in 1996. Its form, which uses a visually even stroke with wide counters, helped signal a new software release. Verdana's members consist of Roman, italic, bold, and bold italic. It has two peers that use nonLatin alphabets: Verdana Greek and Verdana Cyrillic. Verdana is related to Tahoma, a condensed variation of Verdana designed for use in situations that require more information to fit in less space—such as with dialog boxes and menus; and Nina, a spatially efficient sans serif designed for situations that require more information to fit in even less space—such as with small handheld devices. John Berry (2004), who writes and consults extensively on typographic matters, said in *Now Read This* that Meiryo, Verdana's daughter, evolved partially as a response to technologies that enhance photometric resolution permitting more complex writing systems such as scripts to be optimally read on screen and a demand for a Japanese type that weaves the character sets of Kanji, Kana, Katakana and Romanji together fashioning a favorable reading environment for screen and print. Meiryo, too, has Greek and Cyrillic companions. That one family has siblings and offspring, manufactured around a variety of alphabets and writing systems, speaks in part to the economic and political concerns of an international corporation that must respond to the demands of different cultural markets as much as it does to the need for multilingual communication (Fig. 17.14).

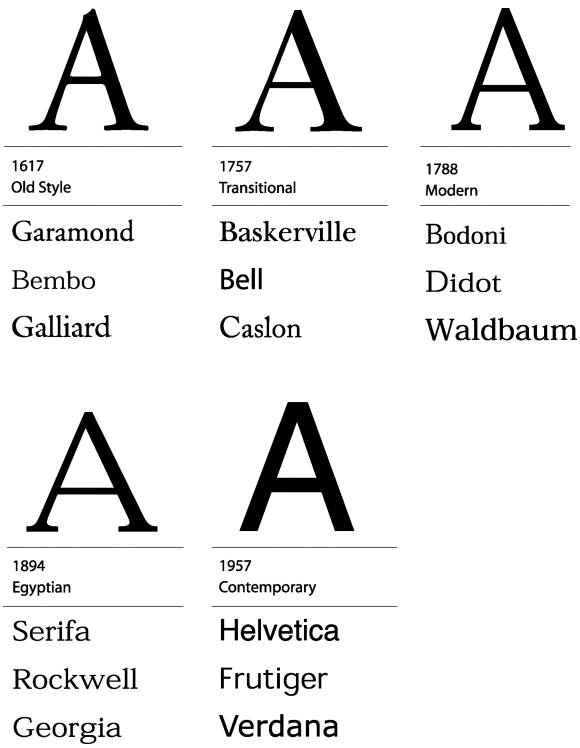


FIGURE 17.14. Type Classification. These five A's are representative of typographic style from the 1600's to present day, reflecting changes in tools, fashion and current events. Ultimately, choice of output media should determine typeface selection, given details such as stress, the thick and thin parts of letterforms, negative space, viewing environment, output resolution, etc.

VARIATIONS IN LETTERFORMS

Variations in Stress

Early type designers mimicked scribal letterforms because they knew and understood these forms. Old-style typefaces have a diagonal stress, a backwards slant—a visual remnant of the pen—that distributes weight through the thin part of the letterform. Over time, the stress traveled several degrees to the right as seen in transitional typefaces. With modern typefaces, the stress becomes rigidly vertical. Egyptians such as Century Schoolbook have a slight diagonal stress; other Egyptians with a stronger slab serif such as Serifa have no stress. Contemporary typefaces have no noticeable stress.

Variation in Thick and Thin

The degree of contrast between the thick and thin strokes of the letters can vary. Old style typefaces have little contrast in strokes. This contrast increases in transitional faces. Extreme stroke contrast is a dominant characteristic of modern typefaces. Egyptian typefaces return to less contrast. Contemporary typefaces have no perceptible thick and thin strokes.

Variations in Serifs

Serifs differ in weight and bracket, which is the shape created by the serif joining the vertical stroke of the letter. Old-style typefaces have heavy concave serifs with thickset brackets. The meticulous serifs of modern typefaces are refined and thin and without brackets. Many Egyptians have heavy straight serifs with little or no bracket. Sans serifs are considered contemporary typefaces (Fig. 17.15).

TYPOGRAPHIC GUIDELINES

Combining Typefaces

Sans serif and serif typefaces can be effectively combined if changes are limited to prevent visual chaos. The key is to ensure that the result respects the content and reinforces the information hierarchy and overall design goals. When combining typefaces,decide whether harmony or contrast is important. Generally, do not use more than two different type families in a document. Remember, at a minimum, a type family offers a Roman, italic, bold, and bold italic. Consider the pattern and texture that the x-heights and stroke weights weave when combined. Excellent typography does not impede the user and the information. Too many typefaces jar and confuse the reader, create visual intrusions, and slow the reading process.

Contrast in Weight (Boldness)

Combining two classic typefaces with a strong differential factor such as Helvetica Extra Bold with Times New Roman can add useful contrast. Be wary of combining intricate typefaces such as Gill Sans Bold and Souvenir, which have structures that may not create compatible reading environments. Too much contrast and visual complexity can be detrimental.

Output Device and Viewing Environment

The quality of publishing technologies and viewing environment vary greatly — laser printer versus video versus electronic

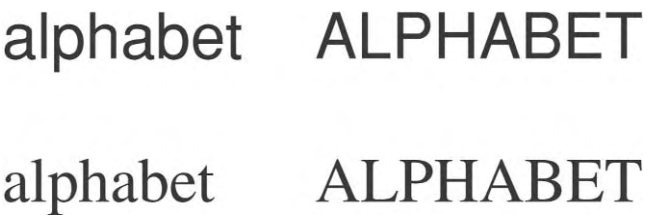


FIGURE 17.15. Serif versus Sans serif. The serif typeface Century above versus the san serif typeface Univers, below.Understanding a typeface's physical characteristics and how it performs in different environments is important. Set a paragraph of text in both a serif and sans serif with exactly the same line length, size and spacing and compare the differences on screen and on paper.

media, etc. In choosing a typeface, its style, size, spacing, and leading, think about the final output medium, and examine this technology's effect on legibility. Low-quality monitors and poor lighting have a major impact: serifs sometimes disappear, letters in small bold type fill in and colored type may disappear altogether.

Letter Spacing and Word Spacing

While the spaces within and between letterforms and words are determined by a type designer in order to set a rhythm that reads well, this spacing can be altered or kerned. However, be careful! When letter spacing is too tight, the letters are hard to distinguish from each other and legibility decreases. When letter spacing is too wide, letter groups are not easily recognized. Spencer argues that optimal letter spacing is inconspicuous, the user can read quickly and easily and understand content. Tight word spacing makes distinguishing individual words difficult. When word spacing is too wide, word groups fall apart. When there is greater space between words than there is between lines, the reader's eye naturally falls to the closest word, which may be below instead of across the line. This frequently occurs with low-resolution or low-cost products.

Line Spacing/Leading

Leading is the distance measured in points between the baseline of one line of text and the baseline of the text line below it. Ascender and descender length influences how closely lines of type can be stacked. The space between lines of text, or leading, should increase in relation to type size. This adjustment is visual not mathematical. Overall legibility may be improved by increasing the leading in relation to column width.

Line Length/Column Width

The correct line length is just long enough for the eye to sweep across without losing its place and easily drop down to continue reading the following lines. A good rule of thumb is that a line of average length contain between 39 and 52 characters.

Justified vs. Ragged Right (FlushLleft)

A justified text column can leave uneven word spacing, creating rivers, or vertical white spaces, within in the paragraph. Rivers cause the eye to move vertically down the page, naturally connecting with what is closest in proximity, instead of moving easily across the line. It is very difficult to prevent rivers in justified text columns without spending considerable effort. Unless the type is manually set or adjusted, which is a time-consuming activity, it's better to use type that is set flush left, ragged right (Fig. 17.16).

A justified column can leave uneven word spacing, creating rivers, or vertical white spaces within the paragraph. These rivers cause the eye to move vertically down the page, to naturally connect visually what is closest in proximity, instead of easily across the line of type. It is very difficult to prevent rivers in justified columns, unless much time and effort is applied. For this reason, unless the type is manually set or adjusted, it is better to use a column set flush left, ragged right.

A justified column can leave uneven word spacing, creating rivers, or vertical white spaces within the paragraph. These rivers cause the eye to move vertically down the page, to naturally connect visually what is closest in proximity, instead of easily across the line of type. It is very difficult to prevent rivers in justified columns, unless much time and effort is applied. For this reason, unless the type is manually set or adjusted, it is better to use a column set flush left, ragged right.

FIGURE 17.16. With current technology, the difference between a justified text column and ragged right text column can make a huge difference in readability. In a poorly justified column, spaces within a justified line connect vertically down the page, distracting the eye from easily reading across a line of text.

Highlighting with Type

Content can be highlighted by modifying type weight, slope, or case. Weight can be shifted from Roman to bold or extra bold. Slope can be altered from Roman to italic. Be mindful that italics are appropriate for short phrases and not long text passages. The italic appears lighter and smaller on the page when compared with its companion Roman and its complex forms are more difficult to read. Case can change from upper and lowercase to all capitals or small caps. Using all caps for extended text passages impedes readability since word outlines are rectangular and harder for the eye to differentiate. Limited shape and size cues are available to help differentiate between letters, words, and sentences to create meaning. Use only one highlighting technique for emphasis.

Decorative Typefaces

Decorative typefaces are of limited use for body text, because their irregular design lessens legibility and should be used in headlines with caution. Because they are essentially typographic fashion statements, decorative typefaces can either reinforce or distract from the overall message or brand of a particular product or organization.

Black on White vs. White on Black and Dark on Light Background vs. Light on a Dark Background

Positive and negative type. White on black (or light on a dark background) is generally regarded as less legible and much more difficult to read over large areas. To the eye, white letters on a black background appear smaller than their reversed equivalent. The amount of contrast between the color of type and the background is an especially important factor for

online communication. Color adds exponential levels of complexity to these considerations since displays are nconsistent from one situation to another (Fig. 17.17 & Fig. 17.18). (Dair, 1967)

DESIGN PRINCIPLES: PAGE DESIGN

Typography deals with legibility and *page design* focuses on readability, the ability to read and comprehend information. Can the user find what is needed on the page? The two important functions of page design are motivation and accessibility. A well-designed page is inviting, drawing the eye into the information. Users are motivated to accept the invitation. An effective page design ensures that the reader continues by increasing the ease of understanding and accessibility of the information. (For purposes of simplicity, the term *page design* is used interchangeably to mean page, screen, and document design.) Motivation and accessibility are accomplished by providing the reader with ways to quickly understand the information hierarchy. At a glance, the page design should reveal easy navigation and clear, intuitive paths to discovering additional details and information. This is called "visual mapping."

A page, site, or product visually mapped for easy navigation has,

- an underlying visual structure or grid, organizational landmarks, graphic cues and other reader aids (Fig. 17.19)
- distinctly differentiated information types
- clearly structured examples, procedures, and reference tools
- well-captioned and annotated diagrams, matrices, charts, and graphics.

This kind of visual structuring helps the reader and provides an obvious path through the materials, aids in skimming, gives a conceptual framework; and, prevents a feeling of information overload.

A table of contents is a simple visual map. It quickly provides a general overview of the order and some details about the structure and content. What it does not reveal, however, are priorities. Site maps or other diagrams provide this type of information as well.

Building the Design of a Page

Effective visual mapping is apparent in the sequence shown below that demonstrates the evolution of an accessible page from plain text. As design elements are added, the page becomes inviting to read and content becomes attainable. The final example organizes the content into units of information using line spacing and vertical thresholds or queues. Differentiation in typeface, weight, and scale reinforce structure. Information design techniques, drawn from cognitive science, can be used to improve communication effectiveness and performance.

Gray page or screen. Raw text interests few readers. When information is presented as a uniform undifferentiated mass, it is difficult and irritating to use and easy to ignore (Fig. 17.20).

Chunking. Structure the visual field by breaking like kinds of information into manageable groups according to subject matter. Chunks in close proximity are read as related. Graphic devices such as rules and line spaces are used to reinforce a grouping and separate chunks (Fig. 17.21).

Queuing. Order information chunks visually to reflect the content hierarchy by addressing the user's requirements of subject matter, order, and importance (Fig. 17.22).

Filtering. Simplify linguistic and visual order by filtering out unnecessary background noise, which interferes with the information being transmitted. Filtering builds a sense of layers of information by using color, visual cues and symbols, and bulleted lists and headers to make a page effective for a range of users and uses (Fig. 17.23).

Mixing modes. People learn through different cognitive modes or styles. Some users favor text, others may prefer illustrations, photos, diagrams, or formulas. To suit these varied learning preferences, information must be translated into several different modes that are then carefully presented to reinforce content and organization (Fig. 17.24).



FIGURE 17.17. Black on White vs. White on Black. The x-height of a typeface (based on the actual height of a lower-case x) is a key characteristic when deciding the visual size of a typeface, particularly when readability is critical. While the above typefaces are the same point size, some seem larger (e.g., Helvetica) and easier to read than others (e.g., Serifa).Letterforms often appear as black/dark shapes on white/light backgrounds. The eye also reads the reverse, or negative shape around a letterform, which can create a shape that visually distracts and makes text difficult to read.Try setting a paragraph of type black on white background, then set the same exact paragraph with white type on a black background. You can also try this same exercise with a dark color on a light background, and then try the reverse. You will notice that the greater the contrast (e.g. white type on black) the harder it is to read in large amounts.

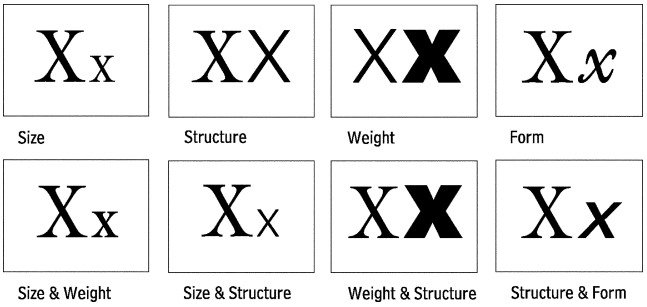


FIGURE 17.18. Typographic Contrasts. This figure which builds on the relationships of contrasts discussed and illustrated in *Design with Type* by Carl Dair (1967), shows typographic contrasts that can be easily used to build information heirarchies.

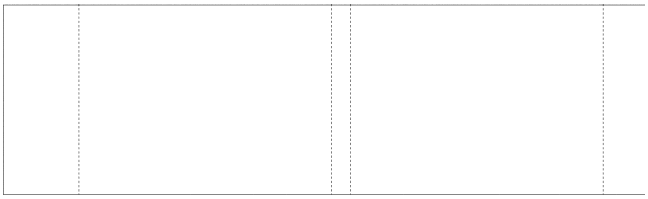
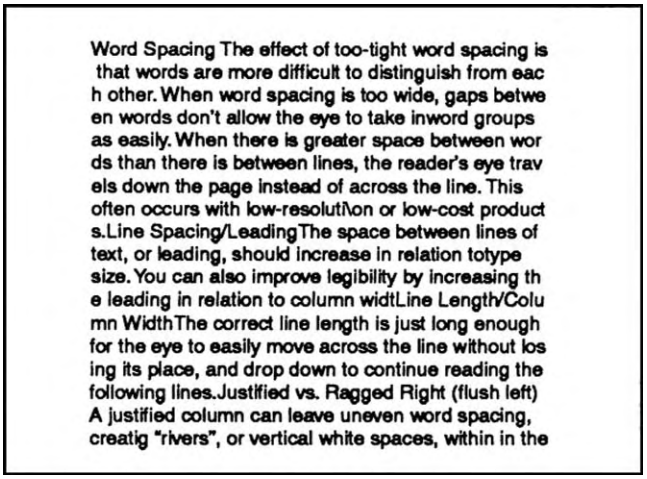
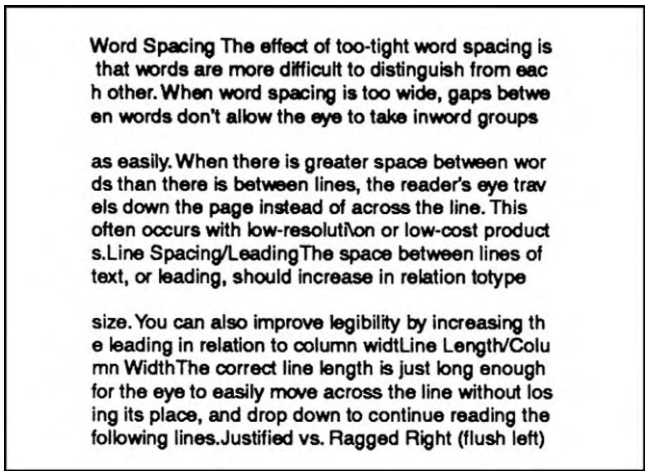


FIGURE 17.19. Grid. The lines that appear this rectangle provide an understanding of what the grid, or underlying structure is of this page. The grid is used as a guide to create more pages that use similar relationships for placement and alignments. It is a point of departure for one who understands the system, to create variations, be a bit more playful, yet still provide a consistent "feel" to the user.



Word Spacing The effect of too-tight word spacing is that words are more difficult to distinguish from eac h other. When word spacing is too wide, gaps betwe en words don't allow the eye to take inword groups as easily. When there is greater space between wor ds than there is between lines, the reader's eye trav els down the page instead of across the line. This often occurs with low-resoluti\on or low-cost product s.Line Spacing/LeadingThe space between lines of text, or leading, should increase in relation totype size. You can also improve legibility by increasing th e leading in relation to column widtLine Length/Colu mn WidthThe correct line length is just long enough for the eye to easily move across the line without los ing its place, and drop down to continue reading the following lines.Justified vs. Ragged Right (flush left) A justified column can leave uneven word spacing, creatig "rivers", or vertical white spaces, within in the

FIGURE 17.20. Gray page or screen.

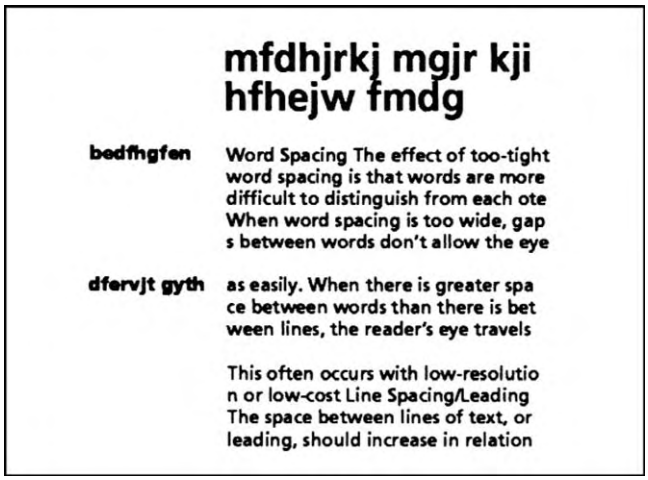


Word Spacing The effect of too-tight word spacing is that words are more difficult to distinguish from eac h other. When word spacing is too wide, gaps betwe en words don't allow the eye to take inword groups

as easily. When there is greater space between wor ds than there is between lines, the reader's eye trav els down the page instead of across the line. This often occurs with low-resoluti\on or low-cost product s.Line Spacing/LeadingThe space between lines of text, or leading, should increase in relation totype

size. You can also improve legibility by increasing th e leading in relation to column widtLine Length/Colu mn WidthThe correct line length is just long enough for the eye to easily move across the line without los ing its place, and drop down to continue reading the following lines.Justified vs. Ragged Right (flush left)

FIGURE 17.21. Chunking.



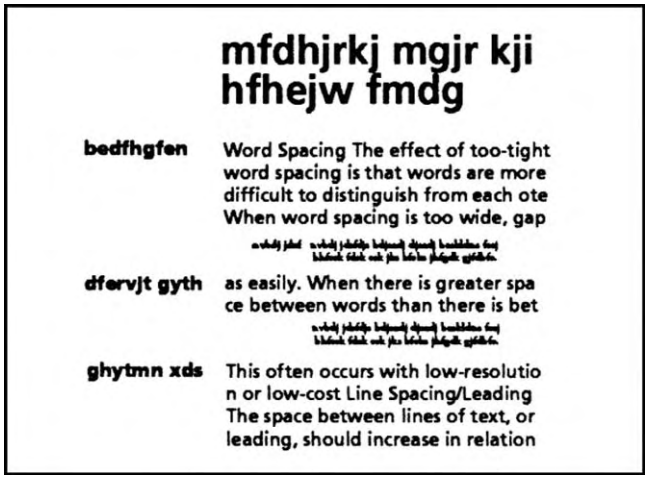
mfdhjrkj mgjr kji hfhejw fmdg

bedfhgfen Word Spacing The effect of too-tight word spacing is that words are more difficult to distinguish from each ote When word spacing is too wide, gap s between words don't allow the eye

dfervjt gyth as easily. When there is greater spa ce between words than there is bet ween lines, the reader's eye travels

This often occurs with low-resolutio n or low-cost Line Spacing/Leading The space between lines of text, or leading, should increase in relation

FIGURE 17.22. Queuing.



mfdhjrkj mgjr kji hfhejw fmdg

bedfhgfen Word Spacing The effect of too-tight word spacing is that words are more difficult to distinguish from each ote When word spacing is too wide, gap

dfervjt gyth as easily. When there is greater spa ce between words than there is bet

ghytmn xds This often occurs with low-resolutio n or low-cost Line Spacing/Leading The space between lines of text, or leading, should increase in relation

FIGURE 17.23. Filtering.



FIGURE 17.24. Mixing Modes.

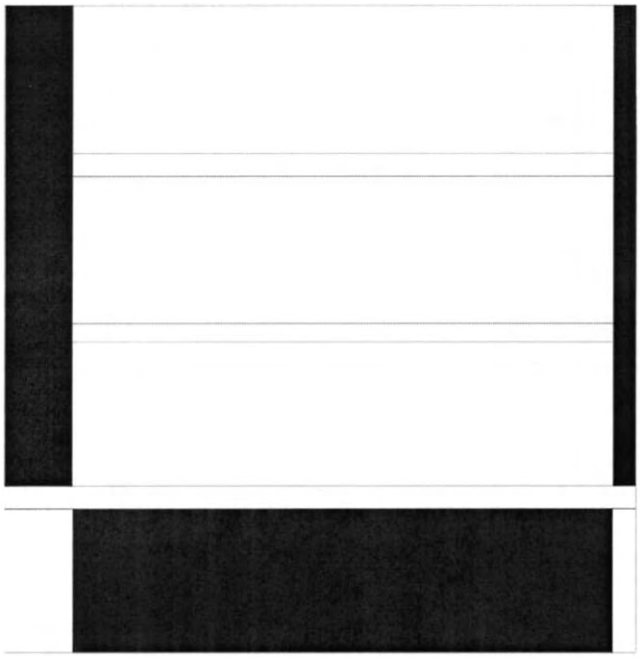


FIGURE 17.25. Abstracting.

Abstracting. The individual page or screen is a microcosm of the complete book, site, or product. The result is a complete codified system of graphic standards, which is effective for both the reader and the producer. Abstracting builds a system of standards that simplifies text organization, creates consistent approaches to preprocessing information, and establishes a customized look for an organization's products (Fig. 17.25).

Other Page Design Techniques

White space. White space (or empty space) is an underutilized but extremely effective design tool. It visually opens up a page, provides focus, helps group like kinds of information, provides resting points for the reader's eye, and creates the perception of simplicity and ease of use.

The grid. A grid is a controlled system of organization that allows for the distribution of visual elements in an intelligible order. A grid, as part of an overall design system, provides an underlying structure that determines the horizontal placement of columns and the vertical placement of headlines, text, graphics, and other artwork.

A grid is built on a series of consistent relationships, alignments, and spatial organizations. It acts as a blueprint that can be repeatedly used to create sequential pages, which are related but respond to different content. When the grid system is understood, it forms the basis for consistent application and extension of the design by others who also understand the intention of the system. Every strong design uses an underlying structure or grid to create a consistent look and feel to any form of visual communication. An analogy can be created between the horizontal and vertical lines that compose a grid used on a page and the metal beams that systematically make up the overall supporting structure of a high-rise building. While the structural supports in a building are consistent from floor to floor, the configuration of the space within each individual floor is based on each occupant's needs. The same hold true with the grid on a page. An important tool that improves usability, a grid enables a user to navigate a page quickly and easily. A grid specifies placement for all visual elements. The user anticipates where a button will appear or how help is accessed. Product or program usefulness and ultimately success are greatly increased through the consistency offered by a grid.

Field of vision. Field of vision refers to what a user can see on a page with little or no eye movement; it is the main area where the eye rests to view most of the page. A good design places key elements in the primary field of vision reflecting and reinforcing the information hierarchy. Size, contrast, grouping, relationships, and movement are tools that create and reinforce field of vision. The user first sees what is visually strongest, not necessarily what is largest or highest. This is particularly true for online information due to the limitations of page real estate and dense information environments.

These concepts, as well as the strength of peripheral vision, can be experienced when viewing a page that has a banner advertisement or moving graphic.

It is virtually impossible to ignore or focus attention on the primary field of vision when there is winking and blinking elsewhere. In fact, superfluous visual devices reduce information's value by distracting and disturbing the user's desire and ability to focus, read and understand.

Proximity. This concept applies to the placement of visual elements physically close to each other so that it is understood that these are related elements.

The illusion of depth. Though the online world exists two-dimensions, contrast can be used to create the illusion of depth. Contrasts created through size, weight, structure, direction, color, texture, and layering can form cues that reinforce hierarchy by giving the illusion that an element is on top of or in front of another.

CHARTS, DIAGRAMS, GRAPHICS, AND ICONS

The goal of any visual device is to provide the fastest, most efficient path to understanding ideas and to make these ideas clear and compelling. Useful, effective graphics act like visual shorthand, particularly important when the real estate of the page is limited (Fig. 17.26). A good visual eliminates the need for text and communicates across cultures. A bad graphic with an unclear meaning that must be reinforced by a long caption can be worse than none at all. The old cliché, a picture is worth a thousand words, is true only if the picture is efficient and effective. In stressful or difficult situations, people do *not* have time to read or the ability to focus on text and/or complex visuals. Though more difficult to achieve, brevity and simplicity in such cases have greater value. Product users prefer well-designed charts, diagrams, and illustrations that quickly and clearly communicate complex ideas and information. Studies show that visual images are retained long after the reader is finished. Designed correctly, visual images can make information memorable and effective. At a minimum, a powerful illustration or graphic can often improve performance simply because it increases user motivation. Visuals are robust communications tools used to (a) visualize and analyze data; (b) present new or abstract concepts; (c) make physical and technical concepts invisible to the eye; and, (d) summarize information efficiently and effectively. Visuals explain and reinforce concepts, relationships and data, making them tangible. Photographs, charts, illustrations, icons, or diagrams become thinking tools. The information is clarified, made easier to evaluate, and has greater impact. Visuals are a very effective way to communicate a message, but choosing the appropriate presentation for a concept is critical to the user's ability to effectively comprehend a message. Understanding the limitations of the display medium is crucial to creating a successful visual.



FIGURE 17.26. Zen calligraphy is an example of the historically close relationship between word and image. The Zen master Hakuin (1768–1865) created this symbol to mean "dead", with additional notes saying "Whenever anyone understands this, then he is out of danger".

Tables, Charts, Diagrams

These three types of graphics are discussed in order of complexity. Tables are the least difficult to create, charts the second most difficult, and diagrams the third. Illustrations, graphics, and other images and visuals are the most complex, require more conceptual and visual sophistication, and may require a consultant to create. When is one more appropriate than the other? Determining which format is the most effective is illustrated in Fig. 17.27. In addition to this list, it is important to remember that visual cues such as color, shading, texture, lines, and boxes should be considered redundant cues and only used to provide additional emphasis to support the concept.

Icons and Visual Cues

Icons and other visual cues are a form of visual shorthand, which helps users locate and remember information. Developing an easily understood style that is consistent with the overall program style is not easy. Choose a style that is simple and consistently reinforced throughout a product. More complex and unique symbols and icons can be used if usage takes place over a longer period, allowing product familiarity and learning to take place. The MasterCard logo consists of two intersecting circles. After many years of reinforcement, most people immediately recognize it without accompanying text or other explanation. It is very difficult to create an icon that, without explanation, communicates a concept across cultures. For example, the use of a freestanding rectangular box with an open door flap indicates mailbox or in-box. This kind of mailbox is rarely used today and was never used in Europe where mail is placed in slots or upright boxes. Even the concept of mail delivery can be considered strange. This is a case where meaning had to be learned. Although simple ideas presented as icons are appropriate, a program with many complex concepts using colloquial images can make using a program agonizing for users from other cultures. There is an important difference between an icon and an illustration, though the two concepts are often confused. If an icon must be labeled, it is really an illustration. The icon's value as visual shorthand is lost. Better to use a word or short phrase rather than word and image when screen space is at a minimum.

A successful icon is memorable with minimal reinforcement. If after viewing an icon several times a user cannot remember its meaning, then the icon is valueless and should be eliminated. Icon sets should share a similarity of style (business-like or playful) and possess formal presentation properties consistent with the overall program or product to which they belong (Fig. 17.28).

When to Use What Graphic

	If you want to show...	use a...
Groups	Group of related items, with a specific order	numbered list
Relationships	Relationships and steps involved in a process	flow chart
	Relationships between categories of ideas	table
	Relationships of tasks taking place over time	project plan table
Evaluate/Compare	Evaluate items against several criteria	rating table
	Evaluate items against one criteria	comparison table
	Compare more than one item to more than one variable	matrix diagram
	Compare several things in relation to one variable	bar chart
	Compare the relative parts that make up a whole	pie chart
Hierarchy	Hierarchical structure of an organization	organizational chart
Concepts	Concept	illustration and/or text icons, other graphics
	Abstract concept	complex images interactive components

FIGURE 17.27.



FIGURE 17.28. AIGA/DOT Icons. These six icons, all from the same set, were developed by the American Institute of Graphic Arts and the U.S. Department of Transportation for use in signage. From upper left to lower right: restaurant, no smoking, trash, stairway leading up, information and taxi.

Illustrations and Photographs

As technology improves, the only limit placed on the use of complex images will be by the designer. The most important consideration is appropriateness of the image for the intended audience. Do not use cartoons for a company brochure, or a low-resolution photograph of a control panel when a line illustration is more effective.

While understanding the meaning and implications of illustrations and photographs is no easy task, there are guidelines for making choices. A photograph can easily represent an existing object, but issues related to resolution and cross-media publishing can make it unintelligible. If a photograph can be reproduced with proper resolution, cropping and contrast, and emphasize a required detail, then photography is a good choice. A photograph can provide orientation and contextual cues that are more difficult to achieve in an illustration. No matter how simplified or cropped to focus attention, a photograph's reproduction quality is often unpredictable. In this situation, a technical illustration such as a line drawing is more effective. An obvious advantage of illustration is that it can present abstract concepts or objects that do not yet exist, or that may never exist. Another benefit is the ability to focus the viewer's attention on detail. For example, a line drawing can place attention on a specific machine part by changing line weight. To achieve a similar result in a photograph adds time, complicates the image, and possibly never simplifies the explanation.

Regardless of the method used to create a visual explanation, it must clarify and reinforce content. If the purpose of the image

is to explain where to locate a piece of equipment, than an overview of the equipment in the environment is appropriate. If the goal is to show an aspect, such as a button location, then the illustration should only focus attention on that aspect. An image can be cropped to focus attention on what is being explained; it depends on the goal of the photograph or illustration.

Situations exist in which it is more effective to use a combination of photography and illustration than either alone. For example, a photograph of an object in its usage environment conveys more information than that of the object itself. If the objective is to show the location of a part of that object, then a line drawing in close proximity to or inset in the photograph is more useful than a photograph or illustration alone).

Guidelines

Visuals should reinforce the message. Don't assume that the audience understands how a visual reinforces the argument (Fig. 17.29). A clear and concise argument must still be made that helps shorten the process of comprehension and learning and causes the user to say, "Aha, that's how it works together!" Visuals should,

- clarify complex ideas
- reinforce concepts
- help the user understand relationships

Create a consistent visual language. Create a consistent visual language that works within the entire communication system. Graphics attract attention. When the user sees a screen, the eye automatically jumps to a visual, regardless of the fact that it may interrupt reading. Graphics should conform to all elements on a page. Unharmonious graphics impede comprehension by increasing the effort needed to understand the relationship between the text and the visual.



FIGURE 17.29. Thirty centuries of development separate the Chinese ancient characters on the left from the modern writing on the right. The meaning of the characters is (from top to bottom): sun, mountain, tree, middle, field, frontier, door.

Consider both function and style. It is important to consider function versus decoration. Albeit wonderful to see an artistically illustrated tax form, is it appropriate to the content or image of both the message and organization it represents? The best graphic is appropriate to the context of the communication and reinforces and validates the message (Fig. 17.30).

Focus on quality vs.quantity. Graphics are only effective if they are carefully planned, executed, and used sparingly. A well-considered diagram with a concise caption is more effective than several poorly thought-out diagrams that require long explanations.

Work with a professional. Many individuals within an organization can write an internal report but a writer or public relations firm is usually commissioned to find the most effective, relevant, and interesting way to communicate a public message such as presented in an annual report or company brochure. Similarly, a designer or visual communications firm should be retained to oversee the development of user interfaces, graph-



FIGURE 17.30. It is obvious which of the above examples communicates an important message most quickly. The goal for the designer is to communicate the message in the most direct way, so that the use can understand and make decisions based on that information. Obviously some situations are more critical than others, but it is no less important to begin design with consideration of the needs of users.

ics, and other visual elements that impact the look and feel and ultimately the overall success of a program.

Build a library to create visual consistency, organizational identity, and a streamlined process. Because graphics require a professional, they can be very time consuming and expensive to create. Once a visual language and style are established, start building a graphics library. If concepts are repeatedly illustrated, streamline the development process by collecting the supporting illustrations and making them available for reuse. An organizational style can be created for these visual explanations. Through repetition, users can learn to associate a style and method of explanation with an organization, which aids in understanding and reinforces product brand and identity.

Reinforce shared meaning (common visual language). A serious issue to consider when creating graphics, particularly conceptual diagrams, is shared meaning, whether it be across an organization or the globe. Individuals can interpret the same diagram in a variety of ways based on backgrounds and experiences.

Truly effective graphics require extra time and effort, but the payoff is tremendous. Graphics are invaluable tools for promoting additional learning and action because they (a) reinforce the message, (b) increase information retention, and, (c) shorten comprehension time.

COLOR

Though color should be considered a reinforcing, or redundant, visual cue, it is by far the most strongly emotional element in visual communication. Color evokes immediate and forceful responses, both emotional and informational. Because color is a shared human experience, it is symbolic. And like fashion, the perception of color changes over time. In all communication, color can be used to trigger certain reactions or define a style. For example, in Western business culture, dark colors such as navy are generally considered conservative, while paler colors such as pink are regarded as feminine. In other cultures, these color choices have an entirely different meaning.

The appropriate use of color can make it easier for users to absorb large amounts of information and differentiate information types and hierarchies. Research on the effects of color in advertising show that ads using one spot of color are noticed 200% more often than black-and-white ads, while full-color ads produce a 500% increase in interest. Color is often used to:

- show qualitative differences
- act as a guide through information
- attract attention/highlight key data
- indicate quantitative changes
- depict physical objects accurately

All in all, color is an immensely powerful tool. Like the tools of typography and page design, it can easily be misused. Research shows that while one color, well used, can increase communication effectiveness, speed, accuracy, and retention, multiple colors when poorly used actually decrease effectiveness. Because it is readily available, it is very tempting to apply color in superficial ways. For color to be effective, it should be used as an integral part of the design program, to reinforce meaning and not simply as decoration. The choice of color—while ultimately based on individual choice–should follow and reinforce content as well as function.

Basic Principles of Color

Additive primaries. The entire spectrum of light is made up of red, green, and blue light, each representing a third of the spectrum. These three colors are known as "additive primaries," and all colors are made up of varying amounts of them. When all three are combined, they produce white light.

Subtractive primaries. If you add and subtract the three primaries, cyan, yellow, and magenta are produced. These are called "subtractive primaries."

Green + Blue – Red = Cyan
Red + Blue – Green = Magenta Red + Green – Blue = Yellow

Color on a computer display is created by using different combinations of red, green, and blue light. In print, colors are created with pigments rather than light. All pigments are made up of varying amounts of the subtractive primaries. The three attributes of color are,

1. ***hue***—the actual color
2. ***saturation***—the intensity of the color
3. ***value***—includes lightness and brightness:

"Lightness" refers to how light or dark a color appears. "Brightness" is often used interchangeably with lightness; however, lightness depends on the color of the object itself, and brightness depends on the amount of light illuminating the object.

How to Use Color

Less is more . . . useful and understandable. Just as you can overload a page or screen with too many typefaces, you can have too many colors. Given the unpredictability of color displays, users, and viewing situations, the choice can get very complicated. Color is often best used to highlight key information. As a general rule, use no more than three colors for primary information. An example is the use of black, red, and gray – black and red for contrasting information, gray for secondary. When thinking about color online, one must remember that each display will output color in a different way. Add to that the lighting situation and a variety of users. All these factors affect color choice.

Create a color logic; use color coding. Use a color scheme that reinforces the hierarchy of information. Don't miscue the audience by using different colors for the same elements. Whenever possible, try to use colors that work with the

project identity or established visual language. Create a color code that is easily understood by the user and reinforces the information.

Create a palette of compatible colors. Harmonious color is created by using a monochromatic color scheme or by using differing intensities of the same hue. However, make them different enough to be easily recognized and simple enough to be easily reproduced, no matter what medium you are using.

Use complementary colors with extreme caution. These are colors that lie opposite each other on the color wheel. Let one dominate and use the other for accents. Never place them next to each because the edges where they meet will vibrate. Though this was the goal of pop art in the 1960s, it makes pages impossible to read. One must check each particular display, as the calibration of monitors can unexpectedly cause this to happen.

Decisions regarding color in typography are critical. Colored type appears smaller to the human eye than the same type in black. This is important to consider when designing user interfaces. One must also consider the "smear" effect on typography in displays, based on the color chosen and interaction with colors around it. Additionally, quality and calibration of displays impact characteristics of color online.

Consider the viewing medium. The same color looks different when viewed on different viewing media such as a computer display, an LCD projector, color laser printer versus dot-matrix output, glossy versus dull paper.

Context is everything. Though printed color is very familiar and more controllable, projected color is inconsistent and varies depending on such things as lighting, size of the color area, size and quantity of colored elements, lighting, and output device. One must check all output/viewing possibilities to insure that a color is readable as well as legible, and not depend on cross-media specification for insuring consistency. What might look good on a laptop may not be readable when projected in a room for hundreds of people to view, and may look completely different when printed in a corporate brochure. The amount of color will affect how it is viewed as well as the best background choice. A blue headline is very readable on a white background, but if that background becomes a color, then readability can be reduced dramatically, depending on how it gets presented on each particular display.

Contrast is critical when making color choices. Contrast is the range of tones between the darkest and the lightest elements, whether one is considering black and white or color. The desired contrast between what is being "read" (this includes graphics, photographs, etc.) must be clearly and easily differentiated from the background it is presented against. If there is not enough contrast (of color, size, resolution, etc.), it will be difficult or impossible to read. This is particularly a problem with online displays, as the designer has no control of quality of the output display.

Quantity affects perception. A small amount of color will be perceived differently than the same color used in a large quantity. In the smaller area, the color will appear darker; in the larger area, the color will appear lighter and brighter.

Use color as a redundant cue when possible. At least 9% of the population, mostly male, is color-deficient to some degree, so it is generally not a good idea to call out warning points only through color. With a combination of color and a different typeface, etc., you won't leave anyone in the dark.

We live in a global world, so when in Rome. . . Remember that different colors have different connotations within various cultures, religions, professions, etc. For example, in the United States on February 14th, red means love, but in Korea, red means death, and in China, red is used in weddings and symbolizes good luck and fortune. In many other countries, red means revolution. To a competitor, red means first place, and to an accountant, red means a negative balance. To a motorist, red means stop, and in emergencies, a red cross means medical help (Fig. 17.31).

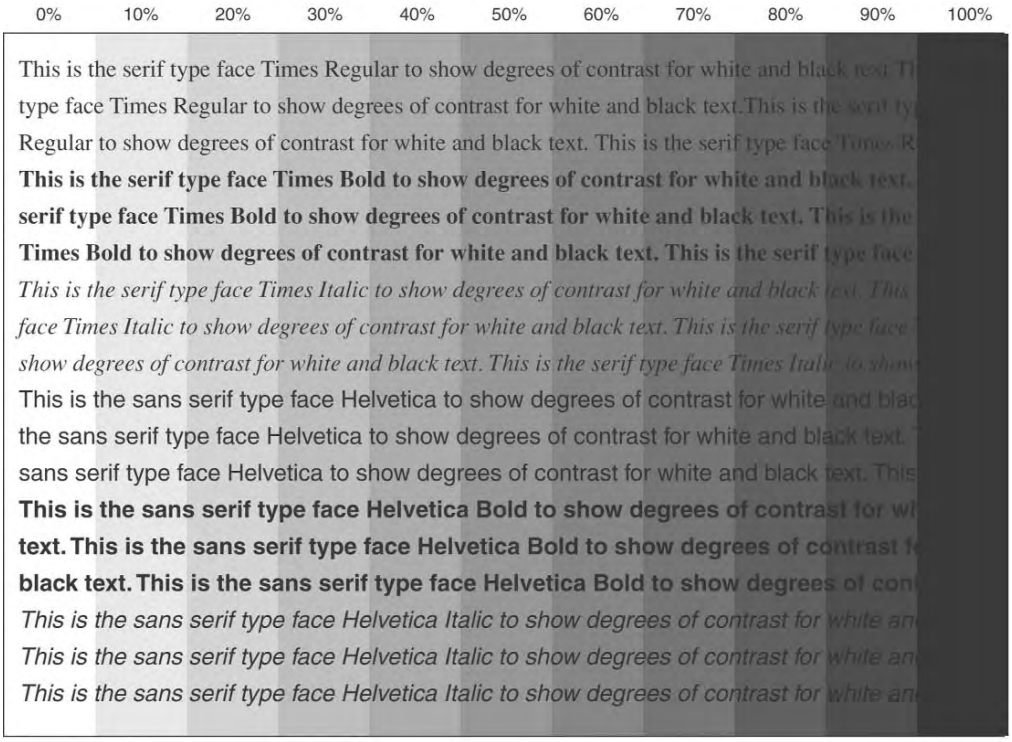
CREATING A SYSTEM: GRAPHIC STANDARDS

With the explosion of new publishing media in a global marketplace, the need for guidelines for developing and producing consistent, quality communication has taken on a new urgency. New technologies make it easy to generate images, offering a wealth of options for experienced and inexperienced publishers alike. The danger lies in creating visual chaos, with every element demanding attention beyond the point of sensory overload. With new tools, chaos can happen faster, at a lower cost, and with greater distribution. A graphic standards manual prevents this confusion.

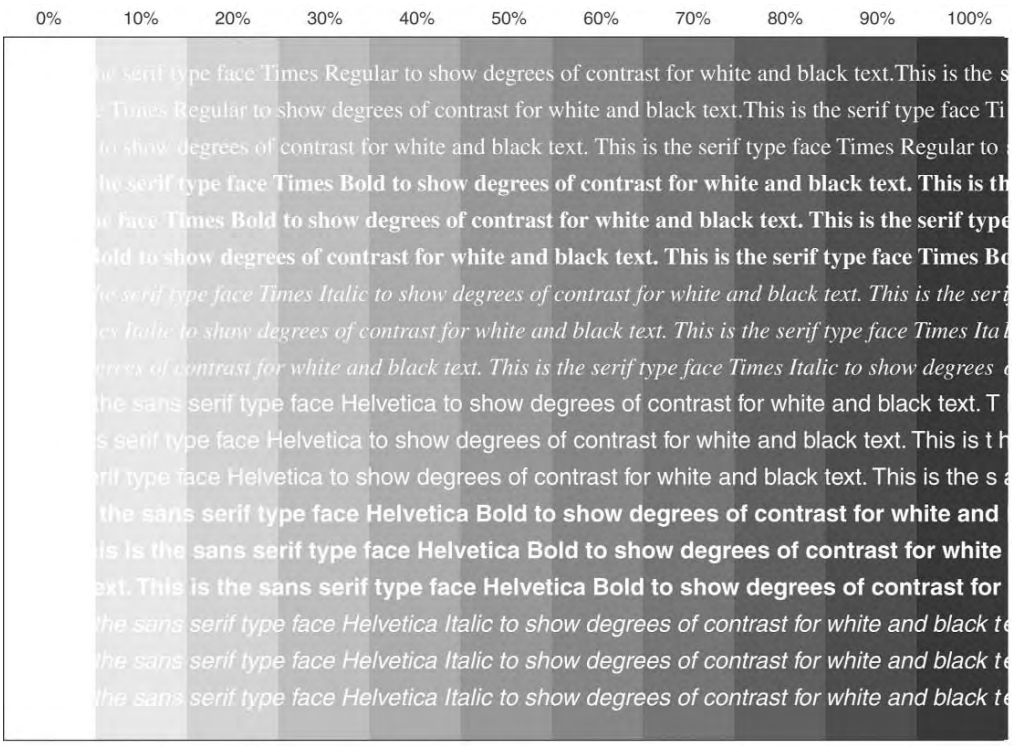
A *graphic standards manual* is the physical manifestation of an identity system. The design historian and educator, Philip Meggs (1998) writes in *A History of Graphic Design* that identity systems arose in the 1950s with the rise of multinational corporations that began to recognize the value and power found in presenting a cohesive visual image globally. A quality-control agent, a standards manual allows an organization to document guidelines and provide tools for organizing and structuring communications and reinforcing brand identity to diverse internal and external audiences. It explains the methodology behind the design, specifies written and visual language, production materials, and methods and gives examples of how to and not to use the identity system so that standards can be implemented by different people, in different places, at different times. A standards manual supports expansion by explaining how to maintain a consistent brand and organizational look and feel as new products, features, and technology are introduced.

A graphic standards system provides,

- Built-in quality: The system ensures that the correct organization/product image is communicated to all audiences. Standards promote consistency in handling information across product lines, divisions, projects, etc.



(a)



(b)

FIGURE 17.31. Trying Examples in Your Context. Since color is not available in this particular edition, try your own experiment. Take a look at this illustration, and recreate a paragraph of text, with the background graded from 100% to 0%, choosing one color for the background. Then set lines of type in a variety of typefaces and sized, to see where it becomes legible or totally impossible to read. The important thing to remember is to test out whatever choices you make within the particular context and parameters, including viewing/projecting devices. Such things as lighting, projection distance, and users' physiological constraints can make all the difference as to whether something can be read or not.

- Control over resources: A system provides dramatic managerial control over resources that use time, money, and materials. Well-developed standards build in flexibility. New communications are easily developed without the original designer and in many instances are developed within the organization itself.
- Streamlined development process: A graphic standards system helps structure thinking for content, design, and production by providing a guideline of predetermined solutions for communication problems. Typical problems are solved in advance or the first time they occur. Most importantly, a graphics standards system encourages the organization as a whole to progress to higher-level issues of communication effectiveness.

What does a system cover? Graphic standards historically have been applied to an organization's logo, stationery, business cards, and other printed materials. As the online portion of an organization's identity dominates, providing for cross-media guidelines is even more critical. Graphic standards are generally communicated to the organization in print and electronic form. Documentation often include,

- Corporate identity manuals: Style guides available in print and online illustrate the application of the standards across the company's publications and provide specifications for production and expansion.
- Templates and guidelines: Templates and guidelines are available in paper and electronic form and are used to develop pages for both environments.
- Editorial style guides: Editorial guides determine the use of product/service names, punctuation, spelling, and writing styles.

Developing the system. When developing a corporate graphics standards system, consider the global publishing needs of the company, the resources available for producing documents, and the skill level of those in charge of production. To responsibly determine overall needs, a team effort is required. Personnel from areas such as information systems, graphic design, usability, and marketing along with engineers, writers, and users should be involved in the process. This team approach helps build support for, and commitment to, the corporate standards. The development of a comprehensive system follows the information design process of audit, development and implementation.

Audit. The audit is a critical step in determining the scope and parameters of an organization's corporate graphic standards. Specific questions for the audit phase include,

- What is the purpose?
- Who are the audiences?
- What are the differences and similarities between audiences?
- Who will do the work?
- How long will it take?
- What tools will be used?
- What is the desired company or product image?

Development. Goals for the development phase include: (a) the design of standards that are easy to read, use, and project a consistent corporate image; (b) design of products that fit within the production parameters of the company.

Implementation. The implementation phase must ensure that the system is accepted and used properly. This requires training and support, easy procedures for distributing and updating materials, and a manual explaining how to use the system. The development of standards is in itself an educational process. It requires all participants to be aware of communication objectives and what is needed to meet them. As alternatives are developed and tested, management has the opportunity to evaluate its company's purpose, nature, and direction as well as its working methods and communication procedures. The process requires commitment and involvement across many departments and levels. The result is an empowering of the organization—planting the seeds for growth and increased effectiveness.

DESIGNING THE EXPERIENCE

The heart of interface design is to define and create the user's experience; what it is really like for people facing the monitor, using a cell phone, or an ATM. Though presentation possibilities are expanding day by day, our capacity to understand, use, and integrate new information and technology has not grown at the same rate. Making the most appropriate media choices, whether image, animation, or sound, to explain complex ideas to widely varied audiences is no easy task. The most important guideline is to understand that there are no rules, only guidelines. It is a generalization to say that a visual principle works a certain way because any change in context changes the application of the principle. For example, in the early days of the software industry, research showed that a specific blue worked well as a background color. Now, however, depending on monitor calibration, as well as environmental lighting, that blue could be a disaster. In fact, that blue can often vibrate if type in particular colors is placed on it. Of course, it depends on the type quantity, size, weight, viewing situation, etc. Sound complex? For this reason, it's important to understand the principles, test the ideas, and then test results on every output devices that will be used. Putting known guidelines together with experience continually gathered from the field allows the designer to develop a clear understanding of what works well in a well-defined environment and user situation. The next key guideline is to keep it simple. Although many tools are available, there is only one goal: to clearly communicate ideas. The designer must always ask, "What is the most efficient and effective way to communicate this idea?" A good illustration might work better (and take less bandwidth) than an animated sequence. Text set in a simple bold headline might allow the user to read the page more efficiently than text placed in a banner moving across the top of a page that constantly draws the eye upward. Animated icons are entertaining, but are they appropriate or necessary for serious financial information? It's tempting to use new tools. The best tip is to use a tool only if it can explain an idea better

than any other tool, enhance an explanation, or illustrate a point that otherwise could not be made as effectively or efficiently. The best design is not noticed; it just works. Products are used to accomplish tasks, *not to draw attention to the design*. The best test of product success is the ease with which a user can understand and complete a task and move on to the next task. Real estate and online real estate are alike in that they both stress location, location, location! With such a premium on space, and so much to accomplish in so little time, be considerate and efficient with online real estate.

Use the elements found in a product's graphic standards appropriately. Constantly consider choices and context and review design principles. The following are issues and considerations to continually evaluate when presenting interactive information.

Effective and appropriate use of the medium. Transitioning a print document to an online environment requires rethinking how the document is presented. Viewing and navigating through online information requires radically different design considerations and methods. Users do not necessarily view the information in a linear way, in a specific order, or timeframe. Interactive media viewed on computer screens have quite different characteristics and potential, particularly as information crosses platforms, resolutions, and environments. The rich medium of print allows a book—a product—to be held, viewed, and read in a sequence determined by the user. The physicality of a book provides sensory cues that are not present on a two-dimensional monitor. Interface designers must find ways to provide equivalent cues that encourage people to handle products comfortably and with confidence.

The element of time. The element of time is the critical difference between static and interactive media. The sense of interaction with a product impacts the user's perception of usefulness and quality. Animated cues such as blinking cursors and other implied structural elements like handles around selected areas become powerful navigational tools if intuitively understood and predictably applied. Consider how the product will be used. Will the user sit down and calmly use the product or will he or she panic and fumble with a keypad? Will the task be completed at one time or at intervals over hours, days, months, and years? The element of time contributes to the design criteria and choices.

CONSISTENT AND APPROPRIATE VISUAL LANGUAGE

A major issue is the unpredictability and vastness of products. Providing way-finding devices that are easy to recognize, understand, and remember, include,

- clear and obvious metaphors
- interface elements consistent with the visual style of other program parts, including consistent style for illustrations, icons, graphic elements, and dingbats.
- guidelines for navigational aids such as color, typography, and page/screen structure that are consistent with other parts of product support.

Navigational Aids

Progress through a book is seen and marked in many ways. Bookmarks and turned corners serve as placeholders. Pens act as mnemonic devices highlighting or underlining text. Table of contents and indices reveal content location. A finger marks a passage to be shared with a colleague as a book is cradled.

Unlike a book, a digital document or program cannot be seen or touched in its entirety. If a document cannot be held, how is specific location known in relation to overall location? How do users return to or move forward through content? How do users travel through unfamiliar space?

Navigational aids provide users with highways, maps, road signs, and landmarks as they move through the online landscape. They enhance discovering and communicate the underlying structure; thereby providing a sense of place so that users know where they are, where they have been, and how to move elsewhere or return to the beginning. Using or building on already familiar visual elements, such as those found in other products and earlier releases, leverages existing knowledge. Graphic standards support this as well. Consistent use of page layout and grid structure makes it easier to remember how information is organized and where it is placed or zoned. This ensures that whatever visual cues are applied can take advantage of the user's experience, ultimately saving time for both the designer and user.

Graphics/Icons

Visual representations such as site maps, graphics, and icons are effective devices for orienting users within a program. A site map offers an overall product view and shows sections or units of information and how these units are related. In the digital environment, graphics and icons assume the role of contents, indexes, and page numbers and can be more effective guides through and around a program than their counterparts in print, because tools such as roll-overs highlight functionality. Creating effective graphics and icons requires that intent and action are defined and designed.

Metaphor

Prior knowledge makes it easier to learn because it provides a conceptual framework on which information can be associated and expanded. When it was first introduced, the metaphor of a desktop with a filing system for a software interface that organized data in a program was easy to grasp because it built on a known experience. Using familiar visual analogies helps users easily understand and organize new information.

Color

Color is a free and very seductive design tool once a monitor is purchased. Use it intelligently. A monitor offers limited workspace. Color can replace or reinforce written explanations when meaning is assigned to it and it is applied methodically. A blue background is always utilized in a testing section and a yellow background in a section overview.

Legibility

Legibility is the ability to read information on the page. The page can be a screen, and as such, has special considerations. Color, size, background, movement, viewing environment, lighting, and resolution play a critical part in legibility.

Readability

Readable screens demand intelligent visual representations and concise, unambiguous text. Meaning can be implied or inferred by the placement of elements in designated areas or zones reserved for distinct information. This makes optimal use of a limited space and increases comprehension and accessibility.

Guidelines

Use the analogy of a poster as a guide to design. A home page is the equivalent of an attention-grabbing poster unpredictably placed in uncontrollable locations. Unlike a home page were a mouse-click on a speaker's name can give biographical information and a click on location can find directions, a poster's static format restricts the amount and depth of information that it can offer.

In print, information is presented in a fixed order. On the web, information is organized hierarchically in a manner that is radically different. Individual users can access the information offered on a website in a sequence that suits their intents and purposes. Online environments offer little regulation over how and in what order the product is accessed. While designers can make suggestions and guesses, this lack of control requires fundamental differences in information presentation. A well-designed home page, like a well-designed poster, should hint at all topics contained in the site, provide high-level information about these topics, and suggest easy paths to access this information. If information goes beyond a single screen, its design must visually communicate location through strong visual hints, so that the user investigates beyond what is immediately visible. Imagine the design considerations required for smaller, handheld, voice-activated devices.

Design for the most difficult common denominator. Design the interface in anticipation of a worst-case scenario. If a manager will use a product in a quiet office with a fast connection, perfect lighting, and a large monitor, then the problem is different from that of a contractor accessing critical information on a laptop in the field. User profile is often unknown because *new technologies define new categories as new opportunities are recognized.* Consider the breadth of possibilities. Design from the user's perspective. Testing, viewing, and questioning can make the difference of product acceptance or not.

Avoid overuse of saturated colors. Saturated colors such as red tend to jump out at the viewer and can be distracting and irritating. Red is usually not a good choice for large areas of on-screen color. High impact is dependent on the contrast between background and foreground colors. On a black background, both yellow and white have a higher impact than red. Consider variations in every viewing situation including how these variations affect contrast among page elements and overall legibility and readability.

Consider different users' levels of skill. Navigational tools should be simple enough for a novice to use, but should not impede an expert. Detailed visual maps and other graphics should be available for those who need them, without hindering an experienced user who wants to bypass an explanation.

Be aware of the fatigue factor. Although there is no definitive answer on fatigue caused by looking at a computer screen for long periods of time, it is a central factor to consider. According to *Color and the Computer,* by H. John Durrett, looking at a well-designed computer screen should not cause any more fatigue than reading a book or writing a report. Though some would disagree with this statement, many people spend more time with their computer than a book and no doubt could offer additional opinions on this subject. As interactive media becomes a commodity, the focus will not be on what a program does, but on how it does it. This will make the difference between product acceptance and product failure. Success or failure will be judged by the ease with which a product is used and how easy users perceive its interface.

Other differences to consider. There are a many differences that impact how and why interfaces are designed, and many of these differences are discussed in more detail in other chapters. A designer should never forget that physical and mental impairments impact an ability to read, comprehend, and use interfaces.

Use the "squint test" to check the design. The squint test is a very simple self-test that checks visual hierarchy. Simply squint at the page so that details are out of focus. What is the first, most dominant element on the page? Is this what should be seen first? What has secondary importance? Cognitive psychology calls this "visual queuing." Successful interaction design creates a visual order that the user can easily follow.

CHALLENGES AND OPPORTUNITIES

Creating Your Own Guidelines

Interactive communication designers face great challenges. How can products that are seen, read, understood, and acted upon be created? Given increasing variety and complexity, how can the power of new technologies be harnessed? How can informed visual choices be made? **WARNING:** No book, seminar, or technology will turn someone into a professional designer! Design is not a craft dependent upon aesthetic ability. Design requires education, training, and experience in a variety of related disciplines equivalent to that of an architect, engineer, surgeon, or cabinetmaker. The following guidelines are offered as starting points, first steps in understanding how to make informed de-

sign decisions—design that provides the best, most thoughtful, and appropriate integration of both form and function.

- **There are no universal rules, only guidelines.** If there were rules, everything would look the same and work perfectly according to these rules. Each situation is different with its own context and parameters.
- **Remember the audience: be a user advocate.** Think about audience needs first throughout the development process. Who is in the audience? What are their requirements? How and where will the audience use the product? The evaluation criteria used in the design-development process springs from the answers to these and other questions. Designers must understand and advocate for the user.
- **Structure the messages.** Analyze content to create a clear visual hierarchy of major and minor elements that reflects the information hierarchy. This visual layering of information helps the user focus on context and priorities.
- ***Test the reading sequence.*** Apply the squint test. How does the eye travel across the page, screen, or publishing medium? What is seen first, second, and third? Does this sequence support the objectives and priorities as defined in the audit?
- **Form follows function.** Be clear about the user and use environment first. An effective interface design represents and reinforces these goals.
- ***Keep things simple.*** Remember the objective is to communicate a message efficiently and effectively, so that users can perform a task. Fewer words, type styles, and graphic elements, mean less visual noise and greater comprehension. An obvious metaphor enhances intuitive understanding and use. The goal is to transfer information, not show off features or graphics.
- **People don't have time to read.** Write clearly and concisely. Design information in an economical, accessible, intuitive format that is enhanced by a combination of graphics and typography. Graphics, if well thought out and designed as an integral part of the page, are very powerful and can efficiently and effectively provide explanations while saving space on a page.
- **Be consistent.** Consistent use of type, page structure and graphic and navigational elements create a visual language that decreases the amount of effort it takes to read and understand a communication piece. The goal is to create a user experience that seems effortless and enjoyable throughout.
- **Start the design process early.** Don't wait. Assemble the development team of designers, usability professionals, engineers, researchers, writers, and user advocates at the beginning of the process. With interactive media, the traditional review and production process will change. The process is less of a hand off and more of a team effort; it's more like making a film than writing a book. Successfully applying the principles of good design enables an organization to communicate more effectively with its audiences and customers improving the worth of its products and services and adding value to its brand and identity.
- **Good design is not about good luck.** Good design for usable interfaces appropriately applies the fundamentals of visual design to interactive products. Creating the most useful, successful design for an interactive product is difficult. The design process is iterative and experiential. There are usually several possible ways to solve a problem, and the final design decision is dictated by the best choices that work within the parameters at any particular time. Advocate on behalf of the user. Users are why designers are here and have this work to do. Users are everywhere, often in places not yet imagined. As the world grows smaller and becomes even more connected, the opportunity lies in where and what has not been discovered.

References

Berry, J. D. (Ed.). (2004). *Now read this: The Microsoft clear type font Collection.* (pp. 15–17, 61–69). Microsoft Corporation.

Bringhurst, R. (2005). *The elements of typographics style* (pp. 17–24). Point Roberts, WA: Hartley & Marks Publishers.

Craig, J. et al. (2006). *Designing with type: The essential guide to typography* (pp. 23–26). New York: Watson-Guptill Publication

Dair, C. (1967). *Design with type* (pp. 49–70). Toronto: University of Toronto Press.

Davis M., Hawley, P., McMullan, B., Spilka, G. (1997). *Design as a catalyst for learning* (pp. 3–12). Alexandria, VA: Association for Supervision and Curriculum Development.

Donis, D. A. (1973a). *A primer of visual literacy (*p. ix). Cambridge, MA: MIT Press.

Donis, D. A. (1973b). *A primer of visual literacy (*pp. 22–23). Cambridge, MA: MIT Press.

Felici, J. (2003). *The complete manual of typography: A guide to setting perfect type* (p. 29). Berkeley, CA: Peachpit Press.

Meggs, P. B. (1998). *A history of graphic design* (p. 363). New York: John Wiley & Son.

Re, M. (Ed.) (2002a). Reading Matthew Carter's Letters. *Typographical speaking: The art of Matthew Carter* (p. 20). Baltimore: Albin O. Kuhn Library & Gallery.

Re, M. (Ed.) (2002b). Reading Matthew Carter's Letters. *Typographical speaking: The art of Matthew Carter* (p. 23). Baltimore: Albin O. Kuhn Library & Gallery.

Re, M. (Ed.) (2002c). Reading Matthew Carter's Letters. "Type Vocabulary." *Typographical speaking: The art of Matthew Carter* (p. 82). Baltimore: Albin O. Kuhn Library & Gallery.

Spencer, H. (1969). *The visible word.* New York: Hastings House.

Tracy, W. (1986). *Letters of credit: A view of type design* (pp. 30–31). Boston, MA: David R. Godine Publisher.

18

GLOBAL/INTERCULTURAL USER INTERFACE DESIGN

Aaron Marcus
Aaron Marcus and Associates, Inc.

Globalization **357**
Definitions of Globalization, Internationalization, Localization 357
Advantages and Disadvantages of Globalization 359
Globalization Development Process 359
Critical Aspects for Globalization: General Guidelines 360
User Demographics 360
Technology 360
Metaphors 360
Mental Models 361
Navigation 361
Interaction 361
Appearance 361
An Example of Specific Guidelines: Appearance 361
Layout and orientation 361
Icons, symbols, and graphics 361
Typography 362
Color 363
Aesthetics 363
Language and Verbal Style 363
Globalization Case Study **363**
Planet Sabre 363
Culture Dimensions **364**
Introduction to Culture Dimensions 364
Culture: An Additional Issue for Globally Oriented UI Designers 365
Hofstede's Dimensions of Culture 366
Power Distance 366
Individualism versus Collectivism 366
Masculinity versus Femininity (MAS) 369
Uncertainty Avoidance (UA) 371
Long- versus Short-Term Time Orientation 373
Design Issues 374
Conclusions and Future Research Issues **376**
Acknowledgments **377**
References **377**

User interfaces (UIs) for successful products and services enable users around the world to access complex data and functions. Solutions to global user interface design consist of partially universal and partially local solutions to the design of metaphors, mental models, navigation, appearance, and interaction. The UI development process must account for many complex localization issues, in particular, the user group's culture. Culture dimensions, *i.e.*, those proposed by anthropologists and other analysts of culture, can provide insight into how designers can/should adjust UIs to better serve users. By managing users' experiences of familiar structures and processes, the UI designer can achieve compelling solutions that enable the UI to be more useful and appealing. Users across the globe will be more productive and engaged with the products and services.

The concept of *User Interfaces for All,* as set forth by Stephanidis (2000), implied the availability of and easy access to computer-based products and services among all peoples in all countries worldwide. Successful computer-based products and services developed for users in different countries and among different cultures (even within one country) consist of partially universal, general solutions and partially unique, local solutions to the design of UIs. Global enterprises seek to mass-distribute products and services with minimal changes to achieve cost-efficient production, maintenance, distribution, and user support. Nevertheless, it is becoming increasingly important, technically viable, and economically necessary to produce localized versions for certain markets. UIs must be designed for specific user groups, not merely translated and given a superficial "local" appearance for quick export to different markets.

Insufficient attention to localization can lead to embarrassing and sometimes critical miscommunication. For example, in Chinese, Coca-Cola means "bite the wax tadpole" or "female horse stuffed with wax," depending on the dialect, which caused the company to change its name to a phonetic equivalent, which means "happiness in the mouth." Similarly, Pepsi's slogan "Come alive with Pepsi" becomes "Pepsi brings your ancestors back from the grave." (Hendrix, 2001). Differences of culture can lead to significant business implications. For example, Saudi Arabia's Higher Committee for Scientific Research and Islamic Law banned Pokemon video games because they "possessed the minds" of Saudi children, thus closing off one of the Middle East's largest markets to the Japanese Nintendo's multibillion-dollar enterprise (Associated Press).

By contrast, attention to localization of language leads to greater comprehension, which can lead to a drop in customer-service costs when instructions are displayed in a user's native language. Moreover, localization can lead to greater attention and retention on the part of viewers/customers. This implication is especially significant for Web-based communication, where Forrester (1998) reported that visitors remain twice as long reviewing local-language sites as they do English-only sites, and business users are three times more likely to buy when communication is in their own language. Developers in the European Union are faced with the daunting but unavoidable challenge of providing websites in more than a dozen languages to succeed in national markets.

By managing the user's experience with familiar structures and processes, surprise at novel approaches, and preferences/expectations, the UI designer can achieve compelling forms that enable the UI to be more usable, useful, and appealing. *Usable* is defined by the international Standards Organization in Switzerland (ISO) as effective, efficient, and satisfying. Globalization of product distribution requires a strategy and tactics for the design process that enables efficient product development, marketing, distribution, and maintenance. Globalization of UI design, whose content and form is so much dependent upon visible languages and effective communication, improves the likelihood that users will be more productive and engaged with computer-based products and services in many different locations globally.

From the designer's perspectives, two primary objectives are (a) provide a consistent UI, and, more generally, user experience across all appropriate products and services, and (b) design products and services with their necessary support systems that are appropriately internationalized (prepared for localization) and localized, i.e., designed for specific markets. Before discussing globalization, localization, and culture issues, we review briefly essential concepts of UI design.

Demographics, experience, education, and roles in organizations of work or leisure characterize users. Their individual needs and wants, hence their goals, as well as their group roles define their tasks. User-centered, task-oriented design methods account for these aspects and facilitate the attainment of effective UI designs that acknowledge and respect users.

UIs conceptually consist of metaphors, mental models, navigation, interaction, and appearance, which may be defined as follows (Marcus, 1995, 1998):

- *Metaphors*: essential concepts conveyed through words and images, or through acoustic, or tactile means. Metaphors concern both over-arching concepts that characterize interaction, as well as individual items, like the "trash can" standing for "deletion" within the "desktop" metaphor.
- *Mental models*: organization of data, functions, tasks, roles, and people in groups at work or play. The term, similar to, but distinct from cognitive models, task models, user models, etc., is intended to convey the organization observed in the UI itself, which is presumably learned and understood by users and which reflects the content to be conveyed, as well as the user tasks.
- *Navigation*: movement through mental models, afforded by windows, menus, dialogue areas, control panels, etc. The term implies dialogue and process, as opposed to structure, i.e., sequences of content potentially accessed by users, as opposed to the static structure of that content.
- *Interaction*: the means by which users communicate input to the system and the feedback supplied by the system. The term implies all aspects of command-control devices (e.g., keyboards, mice, joysticks, microphones), as well as sensory feedback (e.g., changes of state of virtual graphical buttons, auditory displays, and tactile surfaces).
- *Appearance*: verbal, visual, acoustic, and tactile perceptual characteristics of displays. The term implies all aspects of visible, acoustic, and haptic languages (e.g., typography or color; musical timbre or cultural accent within a spoken language; and surface texture or resistance to force).

It is important to bear in mind that localization concerns go well beyond only language translation. They may affect each component of a UI: from choices of metaphorical references, hierarchies in the mental model, and navigation complexity, to choices of input techniques, graphics, colors, sounds/voice/music, and use of vibration.

This chapter discusses the development of UIs intended for users in many different countries with different cultures and languages. The text presents a survey of important issues, as well as recommended steps in the development of UIs for an international and/or intercultural user population and products/services intended for most platforms, including desktop, Web, mobile devices, and vehicles. With the rise of the Internet and application-oriented websites, the challenge of designing good UIs becomes an immediate, practical matter, not only a theoretical issue. This topic is discussed from a user perspective, not a technology or code perspective. The chapter will (a) introduce fundamental definitions of globalization in UI design; (b) demonstrate why globalization is vital to the success of computer-based communication products; (c) introduce culture dimensions; (d) show their effect on one particular platform of desktop Web designs, suggest alternate culture dimensions, and recommend other issues that might relate to culture and user experience.

GLOBALIZATION

Definitions of Globalization, Internationalization, Localization

"Globalization" refers to the entire process of preparing products or services for worldwide production and consumption and includes issues at international, intercultural, and local scales. In an information-oriented society, globalization affects most computer-mediated communication, which, in turn, affects UI design. The discussion that follows refers particularly to UI design.

"Internationalization" refers to the process of preparing code that separates the localizable data and resources (that is, items that pertain to language and culture needed for input and output) from the primary functionality of the software. This separation may include the ability for the UI to work on different platforms in one or more geographic regions where technical, financial, political, and legal matters may affect the designs. Software created in this way does not need to be rewritten or recompiled for each local market. International issues refer to geographic, political, linguistic, and typographic issues of nations or groups of nations. The UIs is the International Standards Organization's (ISO) draft of human factors standards in Europe for color legibility standards of cathode-ray tube (CRT) devices (International Organization for Standardization, 1989) is an example of an effort to establish international standards for some parts of a UI. Other examples are the legal requirement for bilingual English and French displays in Canada, and the quasi-legal denominations for currency, time, and physical measurements, which differ from country to country (Table 18.1).

Intercultural issues refer to the religious, historical, linguistic, aesthetic, and other, more humanistic issues of particular groups or peoples, sometimes crossing national boundaries. Examples (Table 18.2) include calendars that acknowledge various religious time cycles; terminology for color, type, and signs reflecting various popular cultures; and organization of content in Web search criteria reflecting cultural preferences. Sometimes these issues reach national political importance, such as the debate in the redesigned society of Iraq during 2005 over whether Thursday and Friday (the Moslem Sabbath) should constitute the "weekend" rather than Saturday (the Jewish Sabbath) or Sunday (the Christian Sabbath) (Kuhn, 2005).

"Localization" refers to the process of customizing (including language translation but potentially other changes, also, such as content hierarchies, graphics, colors, and icons/symbols) the data and resources of code needed for a specific market. Translation is accomplished manually by in-house staff or one or more outside contractors performing that service. Translation can also be accomplished semi-automatically using software provided by third-party firms, such as Systran (www.Systran.com). Localization can take place for specific, small-scale communities, often with unified language and culture, and usually at a scale smaller than countries, or significant cross-national ethnic "regions." Examples include affinity groups (e.g., French "twenty-somethings," or U.S. Saturn automobile owners), business or social organizations (e.g., German staff of DaimlerChrysler or Japanese golf club members), and specific intranational groups (e.g., India's untouchables, Swedish househusbands, or young Japanese professional women who are rejecting marriage as a life pattern.). Used informally, localization may apply to "corporate cultures" or other groups that may be geographically dispersed. With the spread of Web access, the term "localization" may be used more often to refer to groups of shared interests that may be geographically dispersed. *Note:* this broad definition of "culture" is not accepted by all theorists. For example, see Clausen (2000), as reported by Yardley (2000). However, for the purposes of this chapter, this broad definition is used.

TABLE 18.1. Examples of Differing Displays for Currency, Time, and Physical Measurements

Item	USA Examples	European Examples	Asian Examples
Currency	$1,234.00 (US Dollars)	DM1.234 (German marks)	¥1,234 (Japanese yen)
Time Measures	8:00 P.M., August 24, 1999 8:00 P.M., 8/24/99	20:00, 24 August 1999 (England) 20:00, 24.08.99 (Germany, traditional) 20:00, 1999-08-24 (ISO 8601 Euro standard)	20:00, 1999.08.24, or Imperial Heisei 11, or H11 (Japan)
Physical Measures	3 lb, 14 oz 3′ 10″, 3 feet and 10 inches	3.54 kg, 8.32 m (England) 3,54 kg, 8,32 m (Euro standard)	3.54 kg, 8.32 m in Roman or Katakana chars.

TABLE 18.2. Examples of Differing Displays for Other Data Formats (Partly from Aykin)

Item	USA Examples	European Examples	Asian Examples
Numerics	1,234.56 (also Can., China, UK)	1 234,56 (Finland, Fr, Lux., Portugal., Sweden) 1.234,56 (Albania, Arg,, Denmark, Greece, Neth.) 1'234.56 (Switz: Ger., Ital.) 1'234,56 (Switz: French)	1,234.56
Telephone Numbers	1-234-567-8901, ext. 23 1.234.567.8901 (123) 456-7890	1234 56 78 90 (Austria) (123) 4 5 6 78 90 (Germany) (12) 3456 789 (Italy) +46(0)12 345 67 +49 (1234) 5678-9 (Switz)	+81-53-478-1481 (Japan) 82 2 3142 1100 (Korea) +82-(0)2-535-3893 (Korea) 86 12 34567890 (China)
Address Formats	Title, First Name, MI, Last Name Department Company Number, Street, City, State, Zip-Code Country	Paternal Name, Maternal Name, First Name Company, Department Street, Number City, District/Region Zip-Code, Country (Order may vary from country to country)	Family Name, First Name Department Company Number, Street, Neighborhood, District Zip-Code, City (Japan)

TABLE 18.3. Examples of Differing Cultural References

Item	N. America/Europe Example	Middle-Eastern Example	Asian Example
Sacred colors	White, blue, gold, scarlet (Judeo- Christian)	Green, light blue (Islam)	Saffron yellow (Buddhism)
Reading direction	Left to right	Right to left	Top to bottom
Item	USA	France, Germany	Japan
Web search	"Culture" doesn't imply political discussions	"Culture" implies political Discussions	"Culture" implies tea ceremony Discussions
Sports references	Baseball, football, basketball golf is a sport	Soccer	Sumo wrestling, baseball golf is a religion

Localization changes may need to consider any or all of the following:

Address formats
Alphabetic sequence and nomenclature
Arithmetic operations symbolism
Business standards (quotes, tariffs, contracts, agreement terms, etc.)
Calendar
Character handling
Colors
Content categories
Date and time formats
Documentation nomenclature and formats
Electrical and electronic plug formats and nomenclature
Energy formats
Environmental standards ("green"-compliancy, low energy, low pollution, etc.)
File formats
Font nomenclature, sizes, faces, and byte formats
Frequency (i.e., gigahertz)
Hyphenation and syllabification
Icons and symbols
Intellectual property (protection via patents, copyrights, trademarks)
Keyboard formats
Language differences
Legal processes
Licensing standards
Measurement units (length, volume, weight, electricity, energy, temperature, etc.)
Monetary or currency formats
Multilingual usage
Name formats
Negative formats
Numeric formats and number symbols:
Packaging
Paper formats
Punctuation symbols and usage
Reading/writing direction
Sorting sequences
Style formats
Telephone/fax, temperature formats
Text length
Video recording and playback formats
Voltage/amperage units and formats
Weight formats

Table 18.4 demonstrates the complexities, even within English-language users. Preparing texts in local languages may require use of additional or different characters. The ASCII system, which uses seven or eight bits to represent characters, supports English, and the ISO 8859-1 character set supports Western European languages, including Spanish, French, and German. Other character encoding systems include EBCDIC, Shift-JIS, UTF-8, and UTF-16. ISO has established specific character sets for languages such as Cyrillic, Modern Greek, Hebrew, Japanese, etc. The new Unicode system (ISO 10646) (Graham, 2000) uses 16 bits to represent 65,536 characters, which is sufficient to display Asian languages, like Japanese and Korean, and permits easier translation and presentation of character sets.

Advantages and Disadvantages of Globalization

The business justification for globalization of UIs is complex, but compelling. Clear business reasons can drive decisions to localize content on websites or for Web/mobile applications: Research shows that Web-site visitors stay twice as long at local language sites as they do at English-only sites, business users are three-times more likely to buy when the site is in their own language, and contacts with customer service decline when instructions are shown in the users' native languages (Forrester, 1998).

If the content (functions and data) is likely to be of value to target populations outside of the original market, it is usually worthwhile to plan for international and intercultural factors in developing a product or service, so that it may be customized efficiently, e.g., having separate text files that can be translated more easily. Rarely can a product achieve global acceptance with a "one-size-fits-all" solution. Developing a product for international, intercultural audiences usually involves more than merely a translation of verbal language, however. Visible (or otherwise perceptual, e.g., auditory) language must also be revised, and other UI characteristics may need to be altered.

While increasing initial development costs, developing products or services ready for global use gives rise to potential for increased international sales. However, for some countries, monolithic domestic markets may inhibit awareness of, and incentives for, globalization; e.g., because the United States has in the past been such a large producer and consumer of software, it is not surprising that some U.S.manufacturers have targeted only domestic needs. However, as many U.S. industries, such as movie making and games development, have discovered, foreign sales may be a significant portion of total sales or even larger than domestic sales. In order to penetrate some markets, the local language may be a nearly absolute requirement as in France. Recent reports show that European Union (EU) countries must provide local variations in order to gain user acceptance. As an example: English-only portals are a barrier to EU business success (Vickers, 2000).

Some software products are initiated with international versions (e.g., typically five to seven languages for global products originating in the United States), but they are usually released in sequence because of limited development resources. Other products are "retrofitted" to suit the needs of a particular country, language, or culture, as needs or opportunities arise. In some cases, the later, ad-hoc solution may suffer because of the lack of original planning for globalization.

TABLE 18.4. Comparison of English–Language User Community Conventions

	United States	United Kingdom
Dates	Month/Day/Year: March 17, 2001, 3/17/01	Day/Month/Year: 17 March 2001, 17/03/01
Time	12-hour clock, A.M./P.M. No leading zero (8:32 A.M.)	24-hour clock Leading zero (08:32)
Currency	$189.56, 56¢	GB£189.56, £189.56, 56p
Spelling	Center Color	Centre Colour
Terminology	Truck Bathroom	Lorrie Toilet
Book spine title	Top-down	Bottom up

Globalization Development Process

The "globalized" UI development process is a sequence of partially overlapping steps, some of which are partially, or completely iterative:

- *Plan:* Define the strategy, including the challenges or opportunities for globalization; establish objectives and tactics; determine budget, schedule, tasks, development team, and other resources. Globalization must be specifically accounted for in each item of project planning; otherwise, cost overruns, delays in schedule, and lack of resources are likely to occur. In most cases, business managers will expect to see a return-on-investment (ROI) analysis of the expected benefits, basis for benchmarking, likely tools and process, and metrics to be used in "proving" the results are better.
- *Research:* Investigate dimensions of global variables and techniques for all subsequent steps, e.g., techniques for analysis, criteria for evaluation, media for documentation, etc. In particular, identify items among data and functions that should be targets for change and identify sources of national/cultural/local reference. Globalized user-centered design stresses the need to adequately research users' wants and needs according to a sufficiently varied spectrum of potential users, across specific dimensions of differentiation. Recently, *Fortune*, a major business publication, noted that Microsoft hired a significant number of anthropologists to undertake ethnographic analyses and evaluations in the course of product development (Murphy, 2005). During the past few years, many companies have had anthropologists and ethnographers attend user interface conferences and anthropology conferences to report on their findings. One Internet-based discussion group, Anthropologists in Design (see URL references) regularly tracks inquiries about best information resources, tools, educational institutions, and any case studies related to current job tasks. Local informal gatherings within some urban environments enable designers and anthropologists/ethnographers to get together to exchange experiences.

- *Analyze:* Examine results of challenges or opportunities in the prospective markets, refine criteria for success in solving problems or exploiting opportunities (write marketing or technical requirements), determine key criteria for usability, usefulness, and appeal (i.e., the user experience), and define the design brief, or primary statement of the design's goals. At this stage, globalization targets should be itemized.
- *Design:* Visualize alternative ways to satisfy criteria using alternative prototypes; based on prior or current evaluations, select the design that best satisfies criteria for both general good UI design and globalization requirements; prepare documents that enable consistent, efficient, precise, and accurate implementation.
- *Implement:* Build the design to complete the final product, i.e., write code using appropriate, effective, efficient tools identified in planning and research steps.
- *Evaluate:* At any stage, review or test results in the marketplace against defined criteria for success, i.e., conduct focus groups, test usability on specific functions, gather sales, and user feedback. Identify and evaluate matches and mismatches, and then revise the designs. Test prototypes or final products with international, intercultural, or specific localized user groups to achieve globalized UI designs. As noted in recent proceedings of UI conferences such as the Association for Computing Machinery's Special Interest Group on Human-Computer Interaction (ACM/SIGCHI), Usability Professionals Association (UPA), Human-Computer Interaction International (HCII), and the International Conference on the Internationalization of Products and Services (IWIPS) the techniques of evaluation may need to be adjusted to be successful in different cultures and geographic circumstances (Chavan, 2005). Recent publications have focused specifically on studies of mobile phone users (Lee, Ryu, Smith-Jackson, Shin, Nussbaum, & Tamioka, 2005), as opposed to Web-site visitors.
- *Document:* Record the development history, issues, and decisions in specifications, guidelines, and recommendation documents. Honold (1999) for example, notes that German and Chinese cell-phone users require different strategies for documentation, as well as training, which are related to cultural differences predicted by classical models. Even designing the "ideal" documentation for a particular culture of developers may require user-centered design processes, e.g., those that prefer paper and text, and those that prefer visual, interactive means with minimal text reading.
- *Maintain:* Determine which documents, customer-response services, and other processes will require specialized multiple languages and changes in media or delivery techniques. Prepare appropriate guidelines and templates.
- *Train:* Determine which documents and processes will require multiple languages, different graphics, pacing, media, or distribution methods. Prepare appropriate guidelines and templates.

Critical Aspects for Globalization: General Guidelines

Beyond the UI development process steps identified in the previous section, the following guidelines can assist developers in preparing a "check list" for specific tasks. The following recommendations are grouped under the UI design terms referred to earlier:

User Demographics

- Identify national and cultural target user populations and segments within those populations, and then identify possible needs for differentiation of UI components and the probable cost of delivering them.
- Identify potential savings in development time through the reuse of UI components based on common attributes among user groups. For example, certain primary (or top-level) controls in a Web-based, data-retrieval application might be designed for specific user groups to aid comprehension and to improve appeal. Lower-level controls, on the other hand, might be designed with standardized, unvarying form-like elements.
- Consider legal issues in target communities, which may involve issues of privacy, intellectual property, spamming, defamation, pornography and obscenity, vandalism (e.g., viruses), hate speech, fraud, theft, exploitation and abuse (children, environment, elderly, etc.), legal jurisdiction, seller/buyer protection, etc.

Technology

- Determine the appropriate media for the appropriate target user categories, e.g., emphasis of sound, visual, or three-dimensional (3D) tactile media; verbal versus visual content; etc.
- Account for international differences to support platform, population and software needs, including languages, scripts, fonts, colors, file formats, etc.
- Research and provide appropriate software for code development and content management systems.

Metaphors

- Determine optimum minimum number of concepts, terms, and primary images to meet target user needs.
- Check for hidden miscommunication and misunderstanding.
- Adjust metaphorical images or text to account for national or cultural differences. Publications and projects from/for China and India have suggested metaphors that differ considerably from Western stereotypes. For example, Chavan (1994) stated that Indians relate more easily to bookshelves, books, chapters, sections, and pages, rather than the desktop, file folders, and files. The Wukong protope PDA, developed by Ericsson for Chinese users in 2002, used metaphors based on the Chinese business-social concept of Guang-xi, or relationship maintenance. This approach meant that people, knowledge, and relationships were more fundamental and pervasive than folders, files, and applications (Marcus, 2003).

Mental Models

- Determine optimum minimum varieties of content organization.
- Consider how hierarchies may need to change in detail and overall in terms of breadth and depth. In (Choong & Salvendy, 1999), the authors noted that Chinese and North-American users tended to organize the contents of a house in different ways, and in (Carroll, 1999), the author noted that if one group was given the hierarchies of the other, the group had more difficulty navigating the hierarchy.

Navigation

- Determine need for navigation variations to meet target user requirements, determine cost-benefit, and revise as feasible.

Interaction

- Determine optimum minimum variations of input and feedback. For example, because of Web-access speed differences for users in countries with very slow access, it is important to provide text-only versions, without extensive graphics, as well as alternative text labels to avoid graphics that take considerable time to appear.

Appearance

- Determine optimum minimum variations of visual and verbal attributes. Visual attributes include layout, icons and symbols, choice of graphics, typography, color, and general aesthetics. Verbal attributes include language, formats, and ordering sequences. For example, many Asian written languages, such as Chinese and Japanese, contain symbols with many small strokes. This factor seems to lead to an acceptance of higher visual density of marks in complex public-information displays than is typical for Western countries.

An Example of Specific Guidelines: Appearance

Guidelines for visual and verbal appearance follow. Details can be found in (Aykin, 2004; DelGaldo & Neilson, 1996; Fernandes, 1995; Marcus, 1999; Nielsen, 1990).

Layout and Orientation

- Adjust layout of menus, tables, dialogue boxes, and windows to account for varying reading directions and size of text. Roman languages read only left to right, but Asian languages may read in several directions. For example, in Japan, people read from the right and down, or down and to the right, and Arabic/Hebrew may include right-reading Roman text within left-reading lines of text.
- If areas where dialogue will appear use sentence-like structure with embedded data fields and/or controls, these areas will need special restructuring to account for language changes that significantly alter sentence format. For example, German sentences often have verbs at the ends of sentences, while in English and French verbs are placed in the middle of sentences.
- As appropriate, change layout or imagery that implies or requires a specific reading direction. Left-to-right sequencing may be inappropriate or confusing for use with right-to-left reading scripts and languages. A Web-site design for Arabia On-Line of the late 1990s featured left-to-right English text on a home page intended for Western business people and tourists, but the page layout was still right-to-left, as in Arabic, with primary links at the far right, secondary and tertiary links in the center and left sections of the screen, and small directional arrows pointing right to left. The designers had succumbed to Arabic-language influence to the detriment of English-language readers.
- Check for misleading arrangements of images that lead the viewer's eye in directions inconsistent with language reading directions.
- For references to paper and printing, use appropriate printing formats and sizes. For example, in the United States, standard office letterhead size is 8.5 × 11 inches, but in Europe it is 210mm × 297 mm.

Icons, Symbols, and Graphics

- Avoid the use of text elements and punctuation within icons and symbols to minimize the need for versions to account for varying languages and scripts.
- Adjust the appearance and orientation to account for national or cultural differences. For example, using a mailbox as an icon for e-mail may require different images for different countries.
- Consider using signs or derivatives from international signage systems developed for safety, mass transit, and communication (American Institute of Graphic Arts, 1981; Olgyay, 1995; Pierce, 1996). These signs require little or no translation and may require minimal culture-specific support information.
- Avoid puns and local, unique references that will not transfer well. Note: many "universal" signs may be covered by international trademark and copyright use, e.g., Mickey Mouse and the "Smiley" smiling face.
- Check for appropriateness and use with caution the following:

Animals	People
Body parts and positions	Puns, or plays on words
Colors	National emblems, signs
Hand gestures	Religious, mythological signs

- Consider whether selection symbols, such as the X or check marks, convey correct distinctions of selected versus not-selected items. Some users may interpret an X as crossing out what is not desired, not selection.

- Be aware that office equipment such as telephones, mailboxes, folders, and storage devices differ significantly from nation to nation.

Typography

- Consider character-coding schemes, which differ dramatically for different languages. ASCII is used primarily for English, but single-byte schemes are used for European languages, and double-byte schemes are used for Asian languages. These differences, as well as bidirectional fonts for Hebrew and Arabic display, make it more challenging to support multilingual UIs. Without accounting for character-coding schemes, it is more difficult for users to access content easily.
- Use fonts available for the range of languages required.
- Consider whether special font characters are required for currency, physical measurements, etc.
- Use appropriate alphabetic sequence and nomenclature (e.g., U.S. "zee" versus Canadian/English "zed")
- Ensure appropriate decimal, ordinal, and currency number usage. Formats and positioning of special symbols vary from language to language.
- Consider appropriate numeric formats for decimal numbers and their separators (Aykin, 2000):

1,234.56	Canada, China, United Kingdom, United States
1.234,56	Albania, Argentina, Denmark, Greece, Netherlands
1 234,56	Finland, France, Luxembourg, Portugal, Sweden
1'234.56	Switzerland (German, Italian)
1'234,56	Switzerland (French)

Other numeric issues include the following:

Names of characters

Standards for display of negative and positive numbers

Percent indication

Use of leading zeros for decimal values (e.g., 0.1 or .1)

List separators

Lucky and unlucky numbers (i.e., lucky telephone numbers in Asian countries sometimes sell for higher prices. Note that Chinese prefer even numbers, but Japanese prefer odd numbers.)

- Use appropriate temperature formats, i.e., Fahrenheit, Centigrade, Kelvin
- Use appropriate typography and language for calendar, time zone, and telephone/fax references.
- Consider these date and time issues, among others:

Calendars (Gregorian, Moslem, Jewish, Indian, Chinese, Japanese, etc. Note that even in India there are three major Hindu religious calendars, one used by northern Indians, one by southern Indians, and one shared by both, separate from a secular calendar.)

Character representation (Hindu-Arabic, Arabic, Chinese, Roman, *etc.*)

Clock of 12 or 24 hours

Capitalization rules (e.g., main words in U.S. book titles versus British first only, and the use of capitals for all nouns in German)

Days considered for start of week and for weekend

Format field separators

Maximum and minimum lengths of date and time

Names and abbreviations for days of week and months (two-, three- and multicharacter standards)

Short and long date formats for dates and times

Time zone(s) appropriate for a country and their names

Use of A.M. and P.M. character strings

Use of daylight savings time

Use of leading zeros

- Consider monetary format issues:

Conventions for use with Euros (e.g., double listings)

Credit/debit-card formats, usage conventions

Currency names, denominations

Currency symbols (local versus international versions)

Currency conversion rates

Monetary formats, symbols, and names

Rules for combining different monetary formats: e.g., required multiple currency postings of Euro plus local national currency

Validating monetary input

- Consider name and address formats:

Address elements and number of lines

Address line order

Address: street numbers first or last punctuation:

e.g., 4, route de Monastère; 1504 South 58th Street; Motza Illit 11)

Address: zip/postal codes: alphanumerics,

Address: zip/postal codes: sequence with city, province/state, country

Character sets

Datafield labels (family name versus last name vs surname; first name versus given name versus Christian name)

Field labels (last name versus surname, city/town/district/province, etc.)

Location and location order: neighborhood, district, city/town, state/province)

Name formats:

e.g., family name first for Asian names, family names in all caps in Asia): number of names, number of last names, or surnames (maternal versus paternal), prefixes/titles (e.g., German double title: Dr. Eng.), suffixes (e.g., Jr.)

Name formats:, double family names:

Note, for example, that even within Spanish-speaking countries, some list double family names with maternal first, and others list family names with paternal first. Number of last names/surnames (maternal, paternal)

Number of names (first, middle, initials, etc.)

Prefixes and suffixes

Zip and postal codes (numeric versus alphanumeric, typography, order in relation to city, state/province, or country)

- Consider telephone, fax, and mobile phone-number formats:

  Grouping of digits varies from country to country.

  Internal dialing (initial area-code zeros) versus external (without)

  Numeric versus alphanumeric (i.e., +1-510-767-2676 versus 510-POP-CORN) Number grouping, separators (e.g., (,), +, −, period, space, etc.)

  Use of plus sign for country codes in telephone numbers

  Use of parentheses for area codes

  Format for multiple sequential numbers of businesses (slash, commas, etc.)

Color

- Follow perceptual guidelines for good color usage. For example, use warm colors for advancing elements and cool colors for receding elements; avoid requiring users to recall in short-term memory more than 5 ± 2 different coded colors.
- Respect national and cultural variations in colors, where feasible, for the target users.
- Follow appropriate professional/popular usage of colors, color names, denotation, and connotation.

Aesthetics

- Respect, where feasible, different aesthetic values among target users. For example, some cultures have significant attachment to wooded natural scenes, cartoon characters, textures, patterns, and imagery (e.g., the Finnish and the Japanese), which might be viewed as exotic or inappropriate by other cultures. For example, the Finnish and the Japanese enjoy natural-scenery imagery more than some other countries. The Chinese, Korean, and Japanese seem to enjoy cartoon characters appearing in their products and services more than would seem appropriate for other cultures.
- Consider specific culture-dependent attitudes. For example, Japanese viewers find disembodied body parts, such as eyes and mouths, unappealing in visual imagery.

Language and Verbal Style

- Consider which languages are appropriate for the target users, including the possibility of multiple national languages within one country. For example, English and French within Canada; French, German, and Italian within Switzerland, French or Dutch in Belgium, and in Israel: Hebrew, Arabic, French (official), and English (unofficial). India has more than 20; South Africa has seven, and English is third among native languages. Note also, that some languages have different dialects, i.e., Mexican, Argentinian, and Castillian (Spain) Spanish; Parisian, Swiss, and Canadian French.
- Consider which dialects are appropriate within language groupings and check vocabulary carefully, e.g., for British versus American terms in English, Mexican versus Spanish terms in Spanish, or Mainland China versus Taiwanese terms in Mandarin Chinese.
- Consider the impact of varying languages on the length and layout of text. For example, German, French, and English versions of text generally have increasingly shorter lengths. Some Asian texts are 50%–80% shorter than English; some non-English Roman-character prose texts can be 50% to 200% longer. Some labels can be even longer.

  Example (Aykin, 2000):

English	Undo	Dutch	Ongedaan maken
English	Autoscroll	Swedish	Automatisk rullning
English	Preferences	German	Bildschirmeinstellungen

- Consider the different alphabetic sorting or ordering sequences for the varied languages and scripts that may be necessary and prepare variations that correspond to the alphabets. Note that different languages may place the same letters in different locations, for example, Å comes after A in French but after Z in Finnish.
- Consider differences of hyphenation, insertion-point location, and emphasis, e.g., use of bold, italic, quotes, double quotes, brackets, etc.
- Use appropriate abbreviations for such typical items as dates, time, and physical measurements.
- Remember that different countries have different periods of time for "weekends" and the date on which the week begins.

GLOBALIZATION CASE STUDY

Planet Sabre

An early example of globalization is the UI design for Sabre's Planet Sabre™, one of the world's largest private extranets used by one third of the all travel agents. Sabre contained approximately 42 terabytes of data about airline flights, hotels, and automobile rentals that enabled almost $2 billion of bookings annually, receiving up to one billion "hits" per day. The author's firm worked closely with Sabre over a period of five years to develop the Planet Sabre UI (Marcus, 2001).

The UI development process emphasized achieving global solutions from the beginning of the project. For example, requirements mentioned allowing for the space needs of multiple languages for labels in windows and dialogue boxes. Besides supporting English, Spanish, German, French, Italian, and Portuguese, the UI design proposed switching icons for primary application modules, so they would be more gender-, culture-, and nation-appropriate (see Fig. 18.1 & 18.2).

Figure 18.1 shows the initial screen of Planet SABRE, with icons representing the primary applications or modules within the system conveyed through the metaphor of objects on the surface of a planet. The mail box representing the electronic

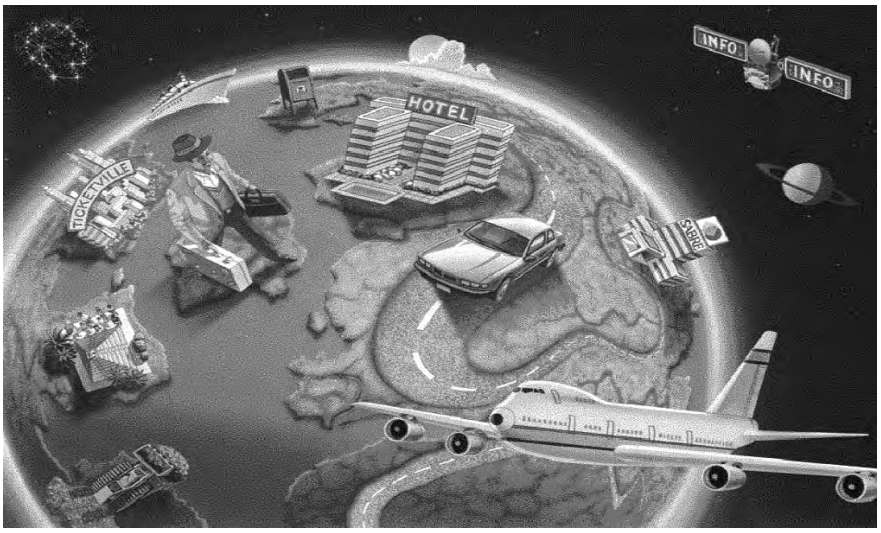


FIGURE 18.1. Example of Planet SABRE home screen showing typical icons for Passenger, Airline Booking, Hotel Rental, Car Rental, and Email (mail box).

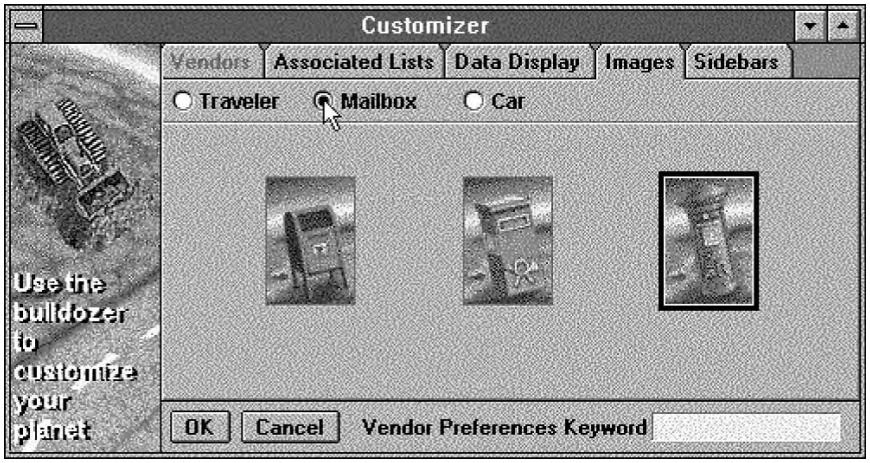


FIGURE 18.2. Example of dialogue box in the Customizer application, by which users can change the icons to become more locally relevant.

mail application depicts an object that users in the USA would recognize immediately; however, users in many other countries might not recognize this object, because mail boxes come in very different physical forms.

Figure 18.2 shows a prototype version of a Customizer dialogue box, in which the user can change preferences; i.e., icons, so they appear throughout the UI with more recognizable images such as the depiction of the passenger.

At every major stage of prototyping, designs developed in the United States were taken to users in international locations for evaluation. User feedback was relayed to the development team and affected later decisions about all aspects of the UI design.

CULTURE DIMENSIONS

Introduction to Culture Dimensions

Localization includes considerations of the target-market cultures. Although relevant for client-server personal-computer software applications and mobile devices, he Web, in particular, enables global distribution of products and services through Internet websites, intranets, and extranets. Professional Web-oriented analysts and designers generally agree that well-designed UIs improve the performance and appeal of websites, helping to convert "tourists" or "browsers" to "residents" and "cus-

tomers." In a global economy, user differences may reflect worldwide culture differences. This section analyzes some of the needs, wants, preferences, and expectations of different cultures. This analysis, as an example, uses a cross-cultural theory developed by Geert Hofstede; however, other competing theories exist, a discussion of which follows.

As an example of culture differences, consider the order in which one prefers to find information. If one is planning a trip by train, is it preferable to see the schedule information first or read about the information provider and assess its credibility? Different cultures look for different data to make decisions.

Culture: An Additional Issue for Globally Oriented UI Designers

Cultures, even within some individual countries, may be very different, e.g., French-Canadians within Canada, or Moslem groups within European countries. As previously noted, sacred colors in the Judeo-Christian West (e.g., red, blue, white, gold) are different from Buddhist saffron yellow or Islamic green. Subdued north-European, Finnish designs for background screen patterns (see Fig. 18.3) might not be equally suitable in south-European, Mediterranean climates. These differences reflect strong cultural values. Batchelor (2000), as reported by Boxer (2001), analyzes the history of Western culture's "fear" of color in a book appropriately titled *Chromophobia*. How might these cultural differences be understood?

Many analysts in organizational communication have studied cultures thoroughly and published classic theories; other authors have applied these theories to analyze the impact of culture on business relations and commerce (e.g., see Elashmawi & Harris 1998; Harris & Moran 1993; Lewis 1991). Culture-oriented works are becoming better known within the UI design community. Anthropological theorists include Geert Hofstede (1997), Edward T. Hall (1969), David Victor (1992), Fons Trompenaars and C. H. Turner (1998), and Shalom Schwarts (2004) each of whom offers valuable insights into the the challenges of cross-cultural communication. Three recent books mention "cultural intelligence" or "cultural awareness" in their titles (Earley, Soon, & Soon, 2003; Kohls, Robert, & Knight, 1994; Thomas & Inkson, 2004).

Other recent publications not only add to the depth of this research, but also generate controversy regarding the permanence and identity of some characterizations of cultures. For example, Nisbett (2003) asserted that essential thinking patterns seem to be culturally influenced and are not universal. He described Western culture's devotion objects, logical reasoning, categorization, and desire to understand situations and events as linear cause-and-effect sequences; while Eastern minds emphasize relationships and seem to accept contradiction more readily.

Clausen (Clausen, 2000), as reported by Yardley (2000), argued that the anthropological term "culture" refers to "the [essentially inescapable] total structure of life of a particular society" while many today use culture to refer to (optional) "shared values." As decribed by Yardley (Yardley, 2000 B3) Clausen argued that ". . . the 'culture' of the Internet has none of the characteristics of a real culture. It is not a total way of life; it did not evolve among a distinct people; nobody inherited it or was raised in it; it makes no moral demands, has no religion at its center, and produces no art." Although complex, its rules are procedural. From the perspective of Internet "dwellers," some of his statements are likely to stir debate, as will his assertions that in terms of the strict definition, culture no longer exists in the United States.

The permanence of cultural attributes is questioned further in articles by Ono and Spindle (2000) and Herskovitz (2000), who note the rise of individualism in classically collectivist Japan and the acceptance, as well as the influence, of Japanese pop-cultural artifacts (i.e., music, movies, and television) in Asian nations that were recently mortal enemies of Japan.

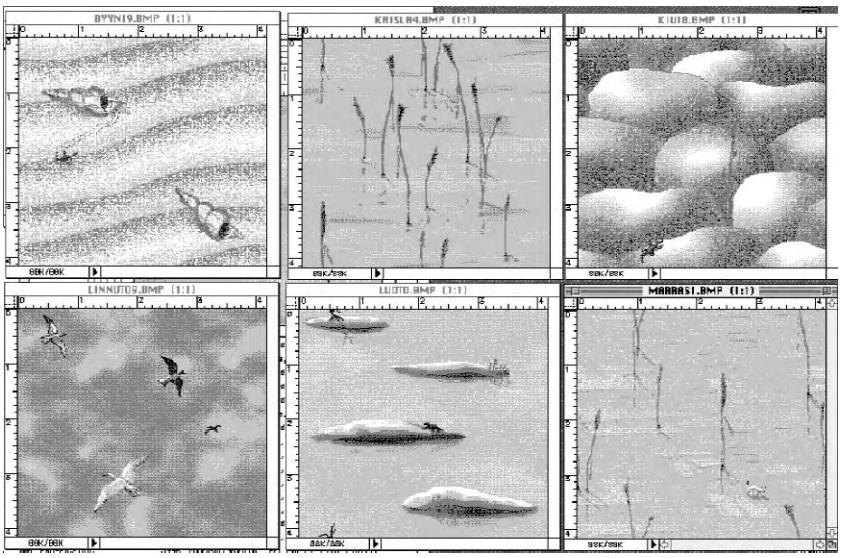


FIGURE 18.3. TeamWare Finnish screen patterns.

Some theorists, including Hofstede (1997), have discussed the relation of culture to economic success and have made differing assertions. Hofstede felt it is a complex mixture of culture plus geography and idiosyncratic drifts in technology, i.e., both culture and creativity. Other theorists argue for solely cultural determinism, as reported by Stille (2001). Still other students of media, sociology, and politics link culture to political and social constructs and discuss the ability of specific groups to manipulate people's cultural self images through media (Lamont, 1995; Brain, 2005).

Although many cultural anthropology theorists are potentially valuable for UI designers, the application of Hofstede's theories will demonstrate the value of this body of research for the fields of UI design, using examples from Web-site design. The following sections introduce Hofstede's concept of cultural dimensions, and applies these dimensions to analyzing Web-site UIs.

Hofstede's Dimensions of Culture

During 1978–1983, the Dutch cultural anthropologist Geert Hofstede conducted detailed interviews with hundreds of IBM employees in 53 countries. Through standard statistical analysis of large data sets, he was able to determine patterns of similarities and differences among their replies. From this analysis, he formulated his theory that world cultures vary along consistent, fundamental dimensions. Because his subjects were constrained to one multinational corporation's worldwide employees, and thus to one company culture, he ascribed their differences to the effects of their national cultures. (One debated characteristic of his approach is that he maintained that each country has just one dominant culture.) In 1997, Hofstede published a version of his research in *Cultures and Organizations: Software of the Mind*. His focus was not on defining culture as refinement of the mind but rather on essential patterns of thinking, feeling, and acting that are well established during childhood. These cultural differences manifest themselves in a culture's choices of symbols, heroes/heroines, rituals, and values.

Hofstede rated 53 countries on indices including five dimensions normalized to values (usually) of 0 to 100. His five dimensions (indices) of culture are,

- Power-distance (PDI)
- Collectivism/individualism (IDV)
- Femininity/masculinity (MAS)
- Uncertainty avoidance (UAI)
- Long/short-term time orientation (LTO)

Each of Hofstede's dimensions follows with an explanation of implications for UI (especially Web UI) design, and illustrations of characteristic websites. The complete data for all countries appears in Table 18.5.

Power Distance

"Power distance" (PD) refers to the extent to which less powerful members expect and accept unequal power distribution within a culture.

High PD countries tend to have centralized political power and exhibit tall hierarchies in organizations with large differences in salary and status. Subordinates may view the boss as a benevolent dictator who expects them to do as they are told. Parents teach children to be obedient and expect respect. Teachers are considered wise and are esteemed. Inequalities are expected, and even may be desired.

Low PD countries tend to view subordinates and supervisors more as equals and more interchangeable, with flatter hierarchies in organizations and less difference in salaries and status. Parents and children, as well as teachers and students, may view themselves more as equals. Equality is expected and generally desired.

Hofstede noted that these differences are hundreds or even thousands of years old. He does not believe they will disappear quickly from traditional cultures, even with powerful global telecommunication systems. Based on this definition, (high versus low) power distance may influence the following aspects of UI design:

- Access to information: highly structured versus not as highly structured.
- Hierarchies in mental models: tall versus shallow.
- Emphasis on the social and moral order (e.g., nationalism or religion) and its symbols: significant/frequent versus minor/infrequent use.
- Focus on expertise, authority, certifications, official logos: strong versus weak.
- Prominence given to leaders versus citizens, customers, or employees.
- Importance of security, restrictions, or barriers to access: explicit, enforced, frequent restrictions on users versus transparent, integrated, implicit freedom to roam.
- Social roles used to organize information (e.g., a managers' section that is obvious to all but sealed off from nonmanagers): frequent versus infrequent.

These PD differences are illustrated by university websites from two countries with very different PD indices (Fig. 18.4 & Fig. 18.5): the Universiti Utara Malaysia (www.uum.edu.my) in Malaysia, with a PD index of 104, the highest in Hofstede's analysis; and and the Technische Universiteit Eindhoven (www.tue.nl) in the Netherlands, with a PD index of 38.

The Malaysian website features strong axial symmetry, a focus on the official seal of the university, photographs of faculty or administration leaders conferring degrees, and monumental buildings in which people play a small role. A top-level menu selection provides a detailed explanation of the symbolism of the official seal and information about the leaders of the university.

The Dutch website, on the other hand, features an emphasis on students (not leaders), an asymmetric layout, and photos of both genders in illustrations. This website emphasizes the power of students as consumers and equals. In similar Dutch university websites, students even have the opportunity to operate a WebCam to take their own tours.

TABLE 18.5. Indices from Hofstede, Geert, *Cultures and Organizations: Software of the Mind*, (Hofstede).

	PDI Rank	Score	IDV Rank	Score	MAS Rank	Score	UAI Rank	Score	LTO Rank	Score
Arab Countries	7	80	26/27	38	23	53	27	68		
Argentina	35/36	49	22/23	46	20/21	56	10/15	86		
Australia	41	36	2	90	16	61	37	51	15	31
Austria	53	11	18	55	2	79	24/25	70		
Bangladesh									11	40
Belgium	20	65	8	75	22	54	5/6	94		
Brazil	14	69	26/27	38	27	49	21/22	76	6	65
Canada	39	39	4/5	80	24	52	41/42	48	20	23
Chile	24/25	63	38	23	46	28	10/15	86		
China									1	118
Columbia	17	67	49	13	11/12	64	20	80		
Costa Rica	42/44	35	46	15	48/49	21	10/15	86		
Denmark	51	18	9	74	50	16	51	23		
East Africa	21/23	64	33/35	27	39	41	36	52		
Equador	8/9	78	52	8	13/14	63	28	67		
Finland	46	33	17	63	47	26	31/32	59		
France	15/16	68	10/11	71	35/36	43	10/15	86		
Germany FR	42/44	35	15	67	9/10	66	29	65	14	31
Great Britain	42/44	35	3	89	9/10	66	47/48	35	18	25
Greece	27/28	60	30	35	18/19	57	1	112		
Guatemala	2/3	95	53	6	43	37	3	101		
Hong Kong	15/16	68	37	25	18/19	57	49/50	29	2	96
India	10/11	77	21	48	20/21	56	45	40	7	61
Indonesia	8/9	78	47/48	14	30/31	46	41/42	48		
Iran	29/30	58	24	41	35/36	43	31/32	59		
Ireland (Rep of)	49	28	12	70	7/8	68	47/48	35		
Israel	52	13	19	54	29	47	19	81		
Italy	34	50	7	76	4/5	70	23	75		
Jamaica	37	45	25	39	7/8	68	52	13		
Japan	33	54	22/23	46	1	95	7	92	4	80
Malaysia	1	104	36	26	25/26	50	46	36		
Mexico	5/6	81	32	30	6	69	18	82		
Netherlands	40	38	4/5	80	51	14	35	53	10	44
New Zealand	50	22	6	79	17	58	39/40	49	16	30
Nigeria									22	16
Norway	47/48	31	13	69	52	8	38	50		
Pakistan	32	55	47/48	14	25/26	50	24/25	70	23	0
Panama	2/3	95	51	11	34	44	10/15	86		
Peru	21/23	64	45	16	37/38	42	9	87		
Philippines	4	94	31	32	11/12	64	44	44	21	19
Poland									13	32
Portugal	24/25	63	33/35	27	45	31	2	104		
Salvador	18/19	66	42	19	40	40	5/6	94		
Singapore	13	74	39/41	20	28	48	53	8	9	48
South Africa	35/36	49	16	65	13/14	63	39/40	49		
South Korea	27/28	60	43	18	41	39	16/17	85	5	75
Spain	31	57	20	51	37/38	42	10/15	86		
Sweden	47/48	31	10/11	71	53	5	49/50	29	12	33
Switzerland	45	34	14	68	4/5	70	33	58		
Taiwan	29/30	58	44	17	32/33	45	26	69	3	87
Thailand	21/23	64	39/41	20	44	34	30	64	8	56
Turkey	18/19	66	28	37	32/3	45	16/17	85		
Uruguay	26	61	29	36	42	38	4	100		
USA	38	40	1	91	15	62	43	46	17	29
Venezuela	5/6	81	50	12	3	73	21/22	76		
West Africa	10/11	77	39/41	20	30/31	46	34	54		
Yugoslavia	12	76	33/35	27	48/49	21	8	88		
Zimbabwe									19	25



FIGURE 18.4. High-power distance: Malaysian University website.

Individualism versus Collectivism

Individualism in cultures implies loose ties; everyone is expected to look after one's self or immediate family but no one else. Collectivism implies that people are integrated from birth into strong, cohesive groups that protect them in exchange for unquestioning loyalty.

Hofstede found individualistic cultures value personal time, freedom, challenge, and such extrinsic motivators as material rewards at work. In family relations, they value being honest, talking things out, using guilt to achieve behavioral goals, and maintaining self-respect. Their societies and governments place individual social-economic interests over the group, maintain strong rights to privacy, nurture strong private opinions (expected from everyone), restrain the power of the state in the economy, emphasize the political power of voters, maintain strong freedom of the press, and profess the ideologies of self-actualization, self-realization, self-government, and freedom.

At work, collectivist cultures value training, physical conditions, skills, and the intrinsic rewards of mastery. In family relations, they value harmony more than honesty/truth (and silence more than speech), use shame to achieve behavioral goals, and strive to maintain face. Their societies and governments place collective social-economic interests over the individual, may invade private life and regulate opinions, favor laws and rights for groups over individuals, dominate the economy, control the press, and profess the ideologies of harmony, consensus, and equality.

Individualism and collectivism may influence, respectively, the following Web UI aspects:

- Motivation based on personal achievement: maximized (expect the extra-ordinary) for individualist cultures versus underplayed (in favor of group achievement) for collectivist cultures.
- Images of success: demonstrated through materialism and consumerism versus achievement of socio-political agendas.
- Rhetorical style: controversial/argumentative speech and tolerance or encouragement of extreme claims versus official slogans and subdued hyperbole and controversy.
- Prominence given to youth and action versus to aged, experienced, wise leaders and states of being.
- Importance of individuals versus products shown by themselves or with groups.
- Underlying sense of social morality: emphasis on truth versus relationships.
- Emphasis on change: what is new and unique versus tradition and history.
- Willingness to provide personal information versus protection of personal data differentiating the individual from the group.

The effects of these differences can be illustrated on the Web by examining national park websites from two countries with very different IC indices (Fig. 18.6 & Fig. 18.7). The Glacier Bay



FIGURE 18.5. Low-power distance: Dutch Educational website.

National Park website (www.nps.gov/glba/evc.htm is located in the United States, which has the highest IC index rating (91). The website from the National Parks of Costa Rica (www .tourism-costarica.com) is located in a country with an IC index rating of 15. The third image (Fig. 18.8) shows a lower level of the Costa Rican website.

Note the differences in the two groups of websites. The U.S. website features an emphasis on the visitor, his or her goals, and possible actions in coming to the park. The Costa Rican website features an emphasis on nature, downplays the individual tourist, and uses a slogan to emphasize a national agenda. An even more significant difference lies below the What's Cool menu. Instead of a typical Western display of new technology or a new experience, the screen is filled with a major political announcement that the Costa Rican government has signed an international agreement against the exploitation of children and adolescents.

Masculinity versus Femininity (MAS)

"Masculinity" and "femininity" refer to gender roles, not physical characteristics. Hofstede focused on the traditional assignment to masculine roles of assertiveness, competition, and



FIGURE 18.6. High individualist (low collectivist) value: US National Park website.



FIGURE 18.7. Low individualist (high collectivist) value: Costa Rican National Park website.

toughness, and the assignment of feminine roles to home and children, people, and tenderness. He acknowledged that, in different cultures, different professions are dominated by different genders. (For example, women dominate the medical profession in the Soviet Union, while men dominate in the United States). However, in masculine cultures, the traditional distinctions are strongly maintained, while feminine cultures tend to collapse the distinctions and overlap gender roles (both men and women can exhibit modesty, tenderness, and a concern with both quality of life and material success.) Traditional masculine work goals include earnings, recognition, advancement, and challenge. Traditional feminine work goals include good relations with supervisors, peers, and subordinates; good living and working conditions; and employment security.

The following list shows some typical masculinity (MAS) index values, where a high value implies a strongly masculine culture:

95 Japan
79 Austria
63 South Africa
62 United States
53 Arab countries
47 Israel
43 France



FIGURE 18.8. What's Cool: Political message about exploitation of children.

39 South Korea
05 Sweden

Since Hofstede's definition focused on the balance between roles and relationships, masculinity and femininity may be expressed on the Web through different emphases. High-masculinity cultures would focus on the following UI design elements:

- Traditional gender/family/age distinctions
- Work tasks, roles, and mastery, with quick results for limited tasks
- Navigation oriented to exploration and control
- Attention gained through games and competitions
- Graphics, sound, and animation used for utilitarian purposes

Feminine cultures would emphasize the following UI design elements,

- Blurring of gender roles
- Mutual cooperation, exchange, and support, (versus mastery and winning)
- Attention gained through poetry, visual aesthetics, appeals to unifying values

Examples of MAS differences on the Web can be illustrated by examining websites from countries with very different MAS indices (Fig. 18.9, Fig. 18.10, & Fig. 18.11). The Woman.Excite website (woman.excite.co.jp) is located in Japan, which has the highest MAS value 95). The website narrowly orients its search portal toward a specific gender, which this company does not do in other countries.

The ChickClick (www.chickclick.com) U.S. website (MAS = 52) consciously promotes the autonomy of young women (leaving out later stages in a woman's life.)

The Excite website (www.excite.com.se) from Sweden, with the lowest MF value 5, makes no distinction in gender or age. (With the exception of the Netherlands, another low MAS country, all other European websites provide more preselected information.)

Uncertainty Avoidance (UA)

People vary in the extent to which they feel anxiety about uncertain or unknown matters, as opposed to the more specific feeling of fear caused by known or understood threats. Cultures vary in their avoidance of uncertainty, creating different rituals and having different values regarding formality, punctuality, legal-religious-social requirements, and tolerance for ambiguity.

Hofstede noted that cultures with high uncertainty avoidance tend to have high rates of suicide, alcoholism, and accidental deaths, and high numbers of prisoners per capita. Businesses may have more formal rules, require longer career commitments, and focus on tactical operations rather than strategy. These cultures tend to be expressive; people talk with their hands, raise their voices, and show emotions. People seem active, emotional, even aggressive. They shun ambiguous situations and expect structure in organizations, institutions, and

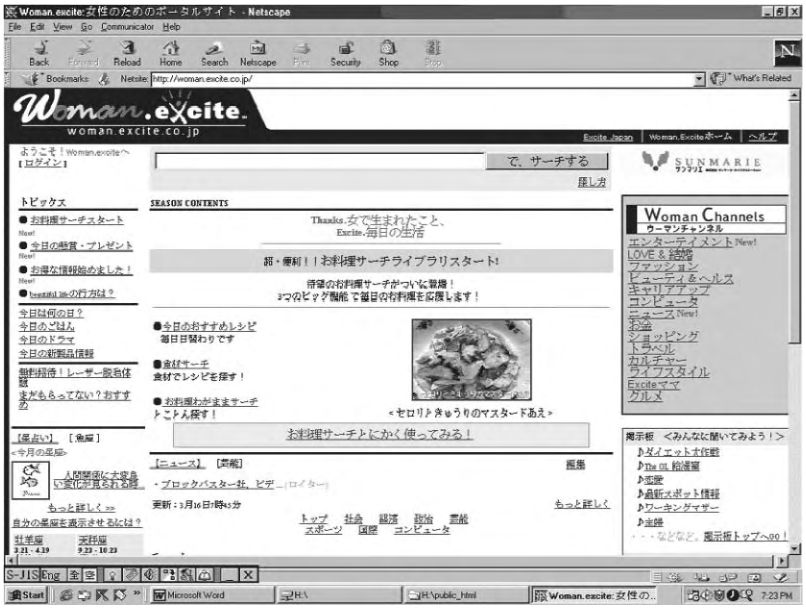


FIGURE 18.9. High masculinity website: Excite.com especially for women in Japan.

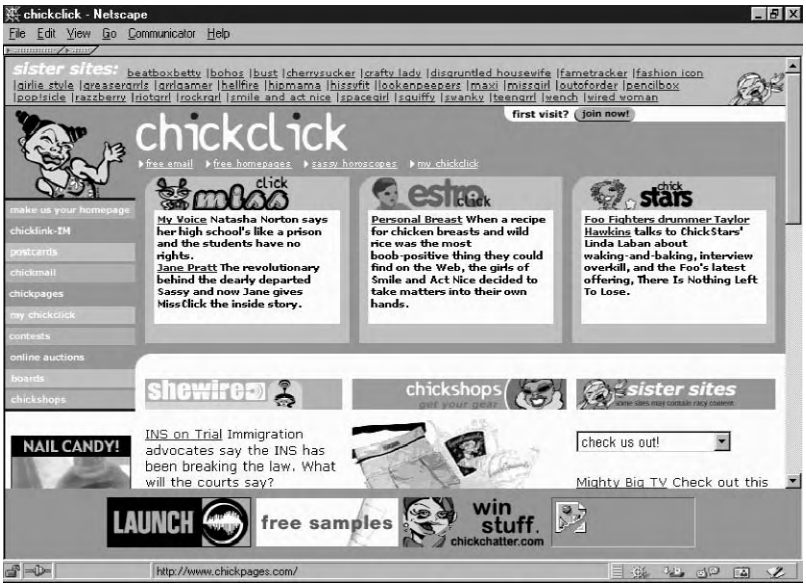


FIGURE 18.10. Medium masculinity website: ChickClick.com in the USA.

relationships to help make events easy to interpret interpret and predictable. Teachers are expected to be experts who know the answers and may speak in cryptic, academic language that excludes novices. In high UA cultures, what is different may be viewed as a threat, and what is "dirty" (unconventional) is often equated with what is dangerous.

By contrast, low UA cultures tend to be less expressive and less openly anxious; people behave quietly without showing aggression or strong emotions (though their excessive caffeine consumption may be to combat depression from their inability to express their feelings.) People seem easygoing, even relaxed. Teachers may not know all the answers (or there may be more than one correct answer), conduct more open-ended classes, and are expected to speak in plain language. In these cultures, what is different may be viewed as simply curious, or perhaps ridiculous.

Based on this definition, uncertainty avoidance may influence contrary aspects of UI and Web design. High-UA cultures would emphasize the following:

- Simplicity, with clear metaphors, limited choices, and restricted data
- Attempts to reveal or forecast results of actions before users act



FIGURE 18.11. Low masculinity website: Swedish Excite.com

- Navigation schemes intended to prevent users from becoming lost
- Mental models and help systems that focus on reducing "user errors"
- Redundant cues (color, typography, sound, etc.) to reduce ambiguity.

Low-UA cultures would emphasize the reverse:

- Complexity with maximal content and choices
- Acceptance (even encouragement) of wandering and risk, with a stigma on "over-protection"
- Less control of navigation; for example, links might open new windows leading away from the original location
- Mental models and help systems might focus on understanding underlying concepts rather than narrow tasks
- Coding of color, typography, and sound maximize information

Online examples of UA differences can be seen on airline websites from two countries with very different UA indices (Fig. 18.12 & Fig. 18.13). The Sabena Airlines website (www.sabena.com) is located in Belgium, a country with a UA index of 94, the highest of the cultures studied. This website shows a home page with very simple, clear imagery and limited choices.

The British Airways website (www.britishairways.com) from the United Kingdom (UA = 35) shows more complexity of content and choices with pop-up windows, multiple types of controls, and "hidden" content that must be displayed by scrolling.

Long- versus Short-Term Time Orientation

In the early 1980s, shortly after Hofstede first formulated his cultural dimensions, he became convinced that a fifth dimension needed to be defined. Long-term time orientation (LT) seemed to play an important role in Asian countries (influenced by Confucian philosophy over thousands of years) that shared these beliefs:

- A stable society requires unequal relations.
- The family is the prototype of all social organizations; consequently, older people (parents) have more authority than younger (and men more than women).
- Virtuous behavior to others means not treating them as one would not like to be treated.
- Virtuous behavior in work means trying to acquire skills and education, working hard, and being frugal, patient, and persevering.

Western countries, by contrast, were more likely to promote equal relationships, emphasize individualism, focus on treating others as you would like to be treated, and find fulfillment through creativity and self-actualization. When Hofstede and Bond developed a survey specifically for Asia and reevaluated earlier data, they found that long-term orientation cancelled out some of the effects of masculinity/femininity and uncertainty avoidance. They concluded that Asian countries are oriented to practice and the search for virtuous behavior while Western countries are oriented to belief and the search for truth. Of the 23 countries compared, the following showed the most extreme values (ranked in parentheses):

118	China (ranked 1)
80	Japan (4)
29	United States (17)
0	Pakistan (23)

High-LT countries would emphasize the following aspects of UI design:

- Content focused on practice and practical value
- Relationships as a source of information and credibility
- Patience in achieving results and goals

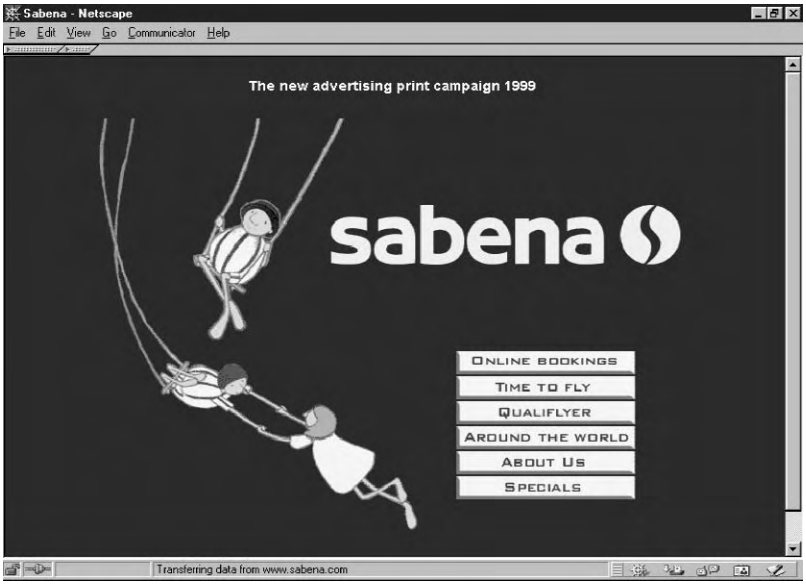


FIGURE 18.12. High uncertainty avoidance: Sabena Airlines website from Belgium.



FIGURE. 18.13. Low uncertainty avoidance: UK's British Airways website.

Low-LT (i.e., short-term time-orientation) countries would emphasize the contrary:

- Content focused on truth and certainty of beliefs
- Rules as a source of information and credibility
- Desire for immediate results and achievement of goals

Online examples of LT differences can be seen in two versions of the same company's website from two countries with different LT values (Fig. 18.14 & Fig. 18.15). The Siemens website (www.siemens.co.de) from Germany (LT of 31) shows a typical Western corporate layout that emphasizes crisp, clean functional design aimed at achieving goals quickly.

The Chinese version from Beijing requires more patience to achieve navigational and functional goals. Typically, such websites also have more emphasis on photos of people and more tolerance of informal, even cartoon-like imagery.

Design Issues

Hofstede noted that some cultural relativism is necessary: it is difficult to establish absolute criteria for what is noble and what

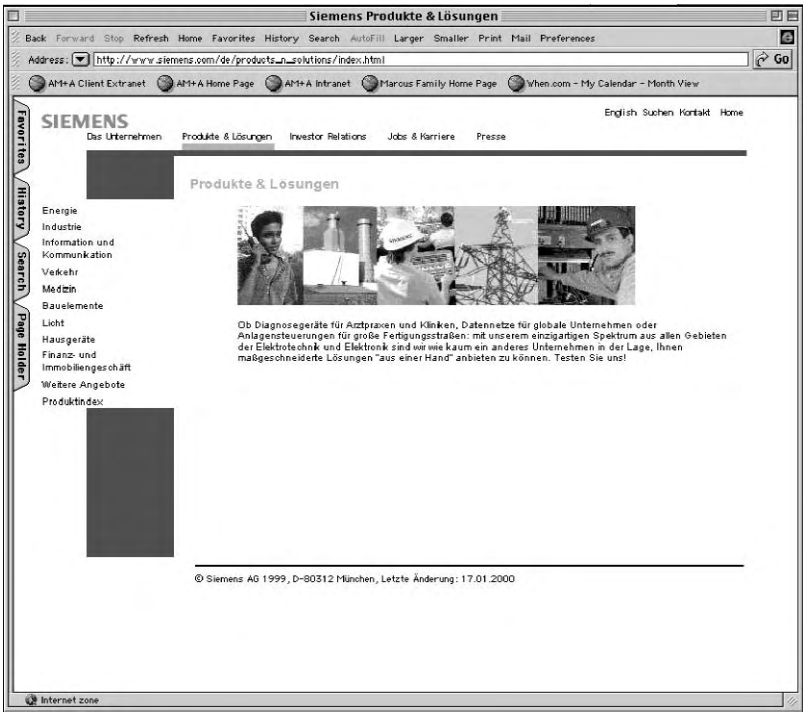


FIGURE 18.14. Low long-term orientation: website form Siemens Germany.



FIGURE 18.15. High long-term orientation. website from Siemens in China.

is disgusting. There is no escaping bias; all people develop cultural values based on their environment and early training as children. Not everyone in a society fits the cultural pattern precisely, but there is enough statistical regularity to identify trends and tendencies. These trends and tendencies should not be treated as defective or used to create negative stereotypes but recognized as different patterns of values and thought. In a multicultural world, it is necessary to cooperate to achieve practical goals without requiring everyone to think, act, and believe identically.

The cultural diffferences just identified seem to survive in websites over periods of time. A study by Marcus and Baumgartner (2003) showed that over a three-year period, the Sabena and British Airways websites cited earlier changed their designs

dramatically. The main air-booking areas had approximately the same number of links, but the areas around this central area differed remarkably. The UK site had twice as many links, indicating again that more complexity was tolerated.

Although this analysis just discussed has used Hofstede's dimensions, many recent publications use other sets of dimensions, often numbering 7±2 dimensions, and sometimes mixing several theorists. Marcus and Baumgartner (2004) published a summary of Baumgartner's thesis, which surveyed approximately 60 experts worldwide regarding their views of the best dimensions from 11 theories through which to evaluate UIs. Twenty-nine dimensions emerged, from which five dimensions seemed to be the highest esteemed: (a) high versus low context, (b) technology attitudes and status, (c) uncertainty avoidance, (d) time perception, and (e) authority perception. While the optimum set of dimensions continues to be debated, more publications within the UI community extol the value of using culture dimensions to analyze and design UIs.

This review of cultural dimensions raises many issues about UI design, especially for the Web:

- How formal or rewarding should interaction be?
- What will motivate different people? Money? Fame? Honor? Achievement?
- How much conflict can people tolerate in content or style of argumentation?
- Should sincerity, harmony, or honesty be used to make appeals?
- What role exists for personal opinion versus group opinion?
- How well are ambiguity and uncertainty avoidance received?
- Will shame or guilt constrain negative behavior?
- What role do community values play in individualist versus collectivist cultures?
- Does the objective of distance learning change what can be learned in individualist versus collectivist cultures? Should these sites focus on tradition? Skills? Expertise? Earning power?
- How should online teachers or trainers act: friends or gurus?
- Would job sites differ for individualist versus collectivist cultures?
- Should there be different sites for men and women in different cultures?
- Would personal Webcams be okay? No okay?
- How much advertising hyperbole is likely to be tolerated?
- Would an emphasis on truth as opposed to practice and virtue require different types of procedural websites for Western or Asian audiences?

If cross-cultural theory becomes an accepted element of UI design, then we may need to change our current practices and tools (e.g., use appropriate testing and focus-group techniques or design-team collaboration techniques suitable for the cultures involved) or develop new ones. We need to make it feasible to develop multiple culturally distinguished versions of websites in a cost-effective manner, perhaps through templates or through specific versioning tools. As personal computer, Web, and mobile products/services continue to develop globally, answering these questions, and exploring, then exploiting, these dimensions of culture, will become a necessity, not an option, for successful theory and practice.

CONCLUSIONS AND FUTURE RESEARCH ISSUES

To achieve culturally sensitive, successful global access to UIs provides many design challenges in the UI-development process. Progress in technology increases the number and kinds of functions, data, platforms, and users of computer-based communication media. The challenge of enabling more people and more kinds of people to use this content and these tools effectively will depend increasingly upon global solutions. By recognizing the need for, and benefit to users of UI designs intended for international and intercultural markets, developers will achieve greater success and increased profitability through the global distribution and increased acceptance of their products.

The recommendations provided in this chapter are an initial set of heuristics that will assist developers in achieving global solutions to their product development. Design methodologies must support globalization throughout the development process. In addition, it is likely that some international and intercultural references will change rapidly, requiring frequent updating of designs. Future work on global UI design may address the following issues:

1. How might global UIs be designed to account for different kinds of intelligence? Gardner (1985) identified the following dimensions of intelligence. These dimensions suggest users might have varying strengths of conceptual competence with regard to using UIs on an individual basis, but these might also vary internationally, or interculturally, due to influences of language, history, or other factors:
   - Verbal/image comprehension
   - Word/image fluency
   - Numerical/graphical fluency
   - Spatial visualization
   - Associative memory
   - Perceptual speed
   - Reasoning
   - Interpersonal awareness
   - Self-awareness
2. How might metaphors, mental models, and navigation be designed precisely for different cultures that differ by such dimensions as age, gender, national or regional group, or profession? The author has posed this issue earlier as a question to the UI analysis/design community (Marcus, 1993, April). The topic is discussed broadly in DelGaldo and Nielson (1996). Further, what means can be developed to enable these variations to be produced in a cost-effective manner using templates?

The taxonomic analyses of global issues for Uis, the theoretical basis for their component selection, the criteria for their

evaluation, and their design methodology have emerged in the UI-development field. Articles about the impact of culture differences on UI design and techniques of ethnographic analysis appear ever more frequently in primary industry publications and in professional conferences such as ACM/SIGCHI, the Usability Professionals Association, the American Anthropologists' Association, IWIPS, and others. The lively exchanges to be found on Anthropologists in Design's Internet discussions likewise attest to the growing numbers of professionals involved in this cross-disciplinary practice (see URL references for resources). Designers should be aware of the scope of the activity, know sources of insight, and incorporate professional techniques in their development process in order to improve the value and success of their international and intercultural computer-mediated products and services.

ACKNOWLEDGMENTS

The author acknowledges Marcus et al. (1999) and Marcus and Gould (2000), on which this chapter is based. Mr. Marcus thanks his staff at Aaron Marcus and Associates, Inc., Sabre, Dr. Emile Gould, Acadia University,, the editors of this book, and Dr. Constantine Stephanidis, the editor of *User Interfaces for All*, for their assistance in preparing this chapter. The author also acknowledges DelGaldo and Nielsen (1996), Fernandes (1995), and Nielsen (1990), which provided a basis for some points raised in this chapter; and the advice of Peter Siemlinger, Dipl.-Ing, Dir., Internat. Inst. for Info. Design, Vienna, Austria, Europe; Prof. Andreas Schneider, Info. Design Dept, Tama Art Univ. Tokyo, Japan; and Dr. Nuray Aykin, New School, New York.

References

American Institute of Graphic Arts (AIGA). (1981). *Symbol Signs*. New York: Visual Communication Books, Hastings House.

Aykin, N. (2000). (personal communication, 12 March 2000).

Aykin, N. (Ed.). (2004). *Usability and Internationalization of Information Technology*. New York: Lawrence Erlbaum Publishers.

Batchelor, D. (2000). *Chromophobia*. London: Reaktion Books.

Boxer, S. (2001, April 28). Vivid color in a world of black and white. *New York Times*, p. A15ff.

Brain, D. (2005). Syllabus for course in sociology of culture at New College, Florida. Retrieved from http://www.ncf.edu/brain/courses/culture/culture_syl05.htm (31 Dec 2005)

Carroll, John M. (1999). "Using Design Rationale to Manage Culture-Bound Metaphors for International User Interfaces, *Proceedings, Internat. Conf. on Internationalization of Products and Services,* Rochester, NY, 20–22 May 1999, pp. 125–131.

Chavan, A. L. (1994). *A design solution project on alternative interface for MS Windows*. Unpublished master's thesis, Royal College of Ave, London, UK.

Chavan, A. L. (2005). Another culture, another method. *Proceedings of the Human Computer Interaction International Conference*, Las Vegas, NV, CD-ROM.

Choong Y., & Salvendy, G. (1999). Implications for design of computer interfaces for Chinese users in mainland China. *International Journal of Human-Computer Interaction, 11*(1), 29–46.

Clausen, Christopher. (2000). *Faded Mosaic: The Emergence of Post-Cultural America*. Chicago, IL: Ivan R. Dee Publisher.

Coriolis Group. (1998). *How to build a successful international website.* Scottsdale, AZ: Author.

Cox, T., Jr. (1994). *Cultural diversity in organizations.* San Francisco: Berrett-Koehler.

Crystal, D. (1987). *The Cambridge encyclopedia of language.* Cambridge, UK: Cambridge University Press.

Daniels, P. T., & Bright, W. (Eds.). (2000). *The world's writing systems.* Sandpoint, ID: MultiLingual Computing.

Day, D. L., del Galdo, E. M., & Prabhu, G. V. (Eds.). (2000, July). Designing for global markets 2. *Proceedings of the 2nd International-Workshop on Internationalisation of Products and Systems*. Rochester, NY: Backhouse Press.

Day, D. (2000, July 13–15). Gauging the extent of internationalization activities. *Proceedings of the 2nd International Workshop on Internationalization of Products and Systems* (pp. 124–136). Baltimore, MD: Backhouse Press.

DelGado, E., & Nielsen, J. (Eds.) (1996). *International User Interface*. New York: Wiley.

Earley, P., Soon, C., & Soon, A. (2003). *Cultural intelligence: Individual interactions across cultures.* Stanford, CA: Stanford University Press.

Elashmawi, F., & Harris, P. R. (1998). *Multicultural management 2000: Essential cultural insights for global business success.* Houston, TX: Gulf.

Evers, V., et al. (Eds.) (2003). *Proceedings of the Fifth International Workshop on Internationalisation of Products and Systems (IWIPS 2003).* Berlin, Germany: Products and Systems Internationalization.

Fernandes, T. (1995). *Global interface design: A guide to designing international user interfaces*. Boston: AP Professional.

Ferraro, G. (2006). *The cultural dimension of international business* (5th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Forrester Research, Inc. (2001, March). *The global user experience*. Cambridge, MA: Forrester Research, Inc.

Gardner, H. (1985). *Frames of mind, the theory of multiple intelligences*. New York: Basic Books.

Goode, E. (2000, August 8). How culture molds habits of thought. *New York Times*, p. D1ff.

Gould, E. W., Zakaria, N., & Shafiz Affendi, M. Y. (2000, 25 September). Applying Culture to website design: A comparison of Malaysian and U.S. websites," *Proceedings of the International Professional Communication Conference, Special interest group on Documentation,* Boston.

Gudykunst, W. B. (2003). *Cross-cultural and intercultural communication.* Thousand Oaks, CA: Sage.

Gudykunst, W. B. (2005). *Theorizing about intercultural communication.*. Thousand Oaks, CA: Sage Publications.

Harris, J., & McCormack, R. (2000). *Translation is not enough.* San Francisco, CA: Sapient.

Harris, P. R., & Moran, R. T. (1993). *Managing cultural differences.* Houston, TX: Gulf.

Hendrix, A. (2001, April 15). The nuance of language. *San Francisco Chronicle*, p. A10.

Herskovitz, J. (2000, December 26). J-Pop takes off: Japanese music, movies, TV shows enthrall Asian nations. *San Francisco Chronicle*, p. C2.

Hofstede, G. (1997). *Cultures and organizations: Software of the mind, intercultural cooperation and its importance for survival.* New York: McGraw-Hill.

Honold, P. (1999). Learning how to use a cellular phone: Comparison between German and Chinese users. *Journal of Society for Technical Communication, 46*, 196–205.

International Organization for Standardization. (1989). *Computer display color* (Draft Standard Document 9241-8). Geneva, Switzerland: Author.

International Standards Organization. (1990). *ISO 7001: Public information symbols*. Geneva, Switzerland: The American National Standards Institute (ANSI).

Kimura, D. (1992). Sex differences in the brain, *Scientific American, 267*(3), September 1992, 118–125.

Konkka, K., & Koppinen, A. (2000). Mobile devices: Exploring cultural differences in separating professional and personal time, *Proceedings of the Second International Workshop on Internationalization of Products and Systems* (pp. 89–104). 13–15 July 2000 Rochester, NY: Backhouse Press.

Lamont, M. (1995). Syllabus for course in contemporary sociological theory, Princeton University. Retrieved from http://www.princeton.edu/~sociolog/grad/courses/fall1995/soc502.html (Retrieved: January 2005)

Lee, Y. S., Ryu, Y. S., Smith-Jackson, T. L., Shin, D. J., Nussbaum, M. A., & Tomioka, K. (2005, July). Usability testing with cultural groups in developing a cell phone navigation system. *Proceedings of the 11th International Conference on Human-Computer Interaction (HCII '05),* Las Vegas, NV. (CD-ROM)

Leventhal, L. (1996). Assessing user interfaces for diverse user groups: Evaluation strategies and defining characteristics. *Behaviour & Information Technology, 15*(3), 127–138.

Lingo Systems. (1999). *The guide to translation and localization* Los Alamitos, CA IEEE Computer Society, ISBN 0-7695-0022-6.

LISA, The Localization Industry Primer. The Localization Industry Standards Association, (LISA) 7, rute du Monastère, 1173 Féchy, Switz., www.lisa.org, 1999, 35 pp.

Marcus, A. (1992). *Graphic design for electronic documents and user interfaces*, Reading MA: Addison-Wesley.

Marcus, A. (1993, April). Human communication issues in advanced UIs, *Communications of the ACM, 36*(4), 101–109.

Marcus, A. (1993, October). Designing for diversity. *Proceedings of the 37th Human Factors and Ergonomics Society,* Seattle, WA, 258–261.

Marcus, A. (1995). Principles of effective visual communication for graphical user interface design. In Baecker, R.; Grudin, J.; Buxton, W.; & Greenberg, S. (Eds.), *Readings in human–computer interaction* (2nd ed., pp. 425–441). Palo Alto, CA: Morgan Kaufman.

Marcus, A. (1998). Metaphor design in user interfaces. *Journal of Computer Documentation, 22*, 43–57.

Marcus, A., Gould, & Chen, E. (1999, June). Globalization of user interface design for the Web. *Proceedings of the 5th Human Factors and the Web Conference,* Gathersburg, MD. (CD-ROM)

Marcus, A. (2001). User interface design for air-travel booking: A case study of Sabre. *Information Design Journal, 10*(2), 186–206.

Marcus, A. (2003). 12 myths of mobile UI design. *Software Development Magazine, May*, 38–40.

Marcus, A. (2005). User interface design and culture. In Aykin, N. (Ed.), *Usability and Internationalization of Information Technology* (pp. 51–78). New York: Lawrence Erlbaum.

Marcus, A., & Baumgartner, V. J. (2004). Mapping user interface design components vs. culture dimensions in corporate websites," *Visible Language*, *38*(1), 1–65.

Marcus, A., & Baumgartner, V. J. (2004). A practical set of culture dimension for evaluating user interface designs. *Proceedings of the Sixth Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004)*, Rotorua, New Zealand, 252–261.

Marcus, A., & Baumgartner, V. J. (2005). A practical set of culture dimensions for global user interface development. In M. Masoodian, S. Jones, B. Rogers, (Eds.), *Proceedings of the Computer Human Interaction: 6th Asia Pacific Conference, APCHI 2004*, Rotorua, New Zealand, 252–261.

Matsumoto, D., Weissmann, M. D., Preston, K., Brown, B. R., & Kupperburd, C. (1997). Context-specific measurement of individualism-collectivism on the individual level: The individualism-collectivism interpersonal assessment inventory. *Journal of Cross-Cultural Psychology, (28)*6, 743–767.

Matsumoto, D., & LeRoux, J. (2003). Measuring the psychological engine of intercultural adjustment: The intercultural adjustment potential scale (ICAPS). *Journal of Intercultural Communications*, 6, pp. 27–52.

Neustupny, J. V. (1987). *Communicating with the Japanese*. The Japan Times, Tokyo, Japan.

Nielsen, J. (Ed.). (1990). *Designing user interfaces for international use: Vol. 13. Advances in human factors/ergonomics*. Amsterdam: Elsevier Science.

Nisbett, R. E. (2003). *The geography of thought: How Asians and Westerners think differently . . . and why.* New York: Free Press.

Olgyay, N. (1995). *Safety symbols art*. New York: Van Nostrand Reinhold.

Ona, Y., & Spindle, B. (2000, December 30). Japan's long decline makes one thing rise: Individualism. *Wall Street Journal*, p. A1.

Ota, Y. (1973). *Locos: Lovers communications system* (in Japanese). Tokyo, Japan: Pictorial Institute.

Ota, Y. (1987). *Pictogram design*. Kashiwashobo Publishers, Tokyo, Japan.

Peng, K. (2000). *Readings in cultural psychology: Theoretical, methodological and empirical developments during the past decade* (1989–1999). New York, NY: John Wiley and Sons.

Perlman, G. (13–15 July). ACM SIGCHI Intercultural Issues. *Proceedings of the Second International Workshop on Internationalization of Products and Systems* (pp. 183–195). Rochester, NY: Backhouse Press.

Prabhu, G. V., & delGaldo, E. M. (Eds.). (1999, 20–22 May). Designing for global markets 1. *Proceedings of the First International Workshop on Internationalization of Products and Systems* (p. 226). Rochester, NY: Backhouse Press.

Prabhu, G., & Harel, D. (1999, August 22–26). GUI design preference validation for Japan and China: A case for KANSEI engineering? *Proceedings of the 8th International Conference on Human-Computer Interaction (HCI International '99),* Munich, Germany.

Saudi Arabia issues edict against Pokemon. (2001, March 27). *San Francisco Chronicle*, p. F2.

Shahar, L., & Kurz, D. (1995). *Border crossings: American interactions with Israelis*. Yarmouth, Maine: Intercultural Press.

Singh, N., & Matsuo, H. (2002). Measuring cultural adaptation on the Web: A content analytic study of U.S. and Japanese websites. *Journal of Business Research, 57(*8), 864–872.

Stille, A. (2001, January 13). An old key to why countries get rich: It's the culture that matters, some argue anew. *New York Times*, 81.

Storti, C. (1994). *Cross-cultural dialogues: 74 brief encounters with cultural difference*. Yarmouth, ME: Intercultural Press.

Tannen, D. (1990). *You just don't understand: Women and men in conversation.* New York: William Morrow and Company, Inc.

Thomas, D., & Inkson, K. (2004). *Cultural intelligence: People skills for global business.* San Francisco: Berrett-Koehler Publishers.

Vickers, B. (2000, November 22). Firms push to get multilingual on the Web. *Wall Street Journal*, p. B11A.

Victor, D. A. (1992). *International business communication.* New York: HarperCollins.

Würtz, E. (2005). A cross-cultural analysis of websites from high-context cultures and low-context cultures. *Journal of Computer-Mediated Communication, 11*(1), article 13.

Yardley, J. (2000, August 7). Faded Mosaic nixes idea of "cultures" in U.S. *San Francisco Examiner*, p. B3.

URLs and Other Information Resources
(Latest version available from author or email to www.AMandA.com)

ACM/SIGCHI Intercultural Issues database: www.acm.org/sigchi/intercultural/
ACM/SIGCHI Intercultural listserve: chi-intercultural@acm.org.
American National Standards Institute (ANSI): www.ansi.org
Anthropologists in Design: http://groups.yahoo.com/group/anthrodesign/
Bibliography of Intercultural publications: www.HCIBib.org//SIGCHI/Intercultural
China National Standards: China Commission for Conformity of Elect. Equip. (CCEE) Secretariat; 2 Shoudu Tiyuguan, NanLu, 100044, P.R. China; Tel: +86-1-8320088, ext. 2659, Fax: +86-1-832-0825
Cultural comparisons: www.culturebank.com, www.webofculture.com, www.iir-ny.com
Digital divide: www.digitaldivide.gov, www.digitaldivide.org, www.digitaldividenetwork.org
Globalization and Internet language statistics: language: www.euromktg.com/globstats/, www.sapient.com, www.worldready.com/biblio.htm
Glossary, six languages: www.bowneglobal.com/bowne.asp?page=9&language=1
International Standards Organization (ISO): http://www.iso.ch/
Internationalization providers: www.basistech.com, www.cij.com, www.Logisoft.com
Internationalization resources: www.world-ready.com/r_intl.htm, www.worldready.com/biblio.htm
Internet users survey, Nua: www.nua.ie/surveys/how_many_online
Japan Info. Processing Society; Kikai Shinko Bldg., No. 3-5-8 Shiba-Koen, Minato-ku, Tokyo 105, Japan; Tel: +81-3-3431-2808, Fax: +81-3-3431-6493
Japanese Industrial Standards Committee (JISC); Min. of Internat. Trade and Industry; 1-3-1, Kasumigaseki, Chiyoda-ku, Tokyo 100, Japan; Tel: +81-3-3501-9295/6, Fax: +81-3-3580-1418
Java Internationalization: http://java.sun.com/docs/books/tutori
Localization Industry Standards Organization (LISA): www.lisa.org
Localization providers: www.Alpnet.com, www.Berlitz.com, www.globalsight.com, www.lhsl.com, www.Lionbridge.com, www.Logisoft.com, www.Logos-usa.com, www.translations.com, www.Uniscape.com
Machine translation providers: www.babelfish.altavista.com, www.IDC.com, www.e-Lingo.com, Lernout & Hauspie <www.lhsl.com>, www.Systransoft.com
Microsoft global development: www.eu.microsoft.com/globaldev/
Simplified English: userlab.com/SE.html
Unicode: www.unicode.org/, www–4ibm.com/software/developer/library/glossaries/unicode.html
World-Wide Web Consortium: www.w3.org/International, www.w3.org/WAI

Recommended Readings

Alvarez, G. M., Kasday, L. R., & Todd, S. (1998). How we made the website international and accessible: A case study. *Proceedings of the 4th Human Factors and the Web Conference*, 5 June 1998, Holmdel, NJ. (CD-ROM)

Day, D., Eves, V., E del Galdo, E. (2005). *Designing for global markets 7: Bridging cultural differences. Proceedings of the International Workshop on Internationalization of Products and Services 2005,* Amsterdam: Grafisch Centrum Amsterdam.

Day, D. L., & Dunckley, L. M. (2001, 12–14 July). *Proceedings of the Third International Workshop on Internationalisation of Products and Systems (IWIPS 2001).* Milton Keynes, UK.

Doi, T. (1973). *The anatomy of dependence*. New York: Kodansha International.

Doi, T. (1986). *The anatomy of self: The individual versus society*. New York: Kodansha International.

Dreyfuss, H. (1966). *Symbol sourcebook*. New York: Van Nostrand Rhinehold.

Fetterman, D. M. (1998). *Ethnography: Step by step* (2nd ed.). Thousand Oaks, CA: Sage.

Forrester Research, Inc. (1998). *JIT Web localization.* Retrieved 4 July 1998, from www.Forrester.com

French, T., & Smith, A. (2000, 13–15 July). Semiotically enhanced Web interfaces: Can semiotics help meet the challenge of cross- cultural design? Proceedings of the 2nd International Workshop on Internationalization of Products and Systems (pp. 23–38). Baltimore, MD. Rochester, NY: Backhouse Press.

Graham, T. (2000). *Unicode: A primer.* Sandpoint, ID: MultiLingual Computing and Technology.

Hall, E. T. (1969). *The bidden dimension.* New York: Doubleday.

Harel, D., & Girish, P. (1999, 22–23 May). Global User Experience (GLUE), Design for cultural diversity: Japan, China, and India. *Designing for Global Markets*, *Proceedings of the First International Workshop on Internationalization of Products and Systems (IWIPS-99)* (pp. 205–216). Rochester, NY: Backhouse Press.

Hoft, N. L. (1995). *International technical communication: How to export information about high technology*. New York: John Wiley and Sons, Inc.

Inglehart, R. F., Basanez, M., & Moreno, A. (Eds.) (1998). *Human values and beliefs: A cross-cultural sourcebook.* Ann Arbor, MI: University of Michigan Press.

International Standards Organization. (1993). *ISO 7001: Public information symbols: Amendment 1*. Geneva, Switzerland: The American National Standards Institute (ANSI).

Kohls, L. R., & Knight, J. M. (1994). *Developing intercultural awareness: A cross-cultural training handbook* (2nd ed.). Yarmouth: ME: Intercultural Press.

Kuhn, A. (2005, March 12). Mulling weekends, workdays in Iraq. *National Public Radio Web Archive*. Retrieved January 12, 2006, from http://www.npr.org/templates/story/story.php?storyId=4540715

Kurosu, M. (1997, August 24–29). Dilemma of usability engineering. In G. Salvendy, M. Smith, & R. Koubek (Eds.), Design of computing systems: Social and ergonomics considerations, Volume 2. *Procedings of the 7th International Conference on Human-Computer Interaction HCI International '97* (pp. 555–558). Amsterdam: Elsevier.

Lewis, R. (1991). *When cultures collide*. London: Nicholas Brealey.

Marcus, A. (2000). International and intercultural user interfaces. In Dr. C. Stephanidis (Ed.), *User Interfaces for All*. (pp. 47–63). New York: Lawrence Erlbaum Associates.

Marcus, A., & Gould, E. W. (2000). Crosscurrents: Cultural dimensions and global web user interface design. *Interactions, 7*, 32–46.

Murphy, R. (2005). Getting to know you. *Fortune Small Business*. Retrieved November 6, 2005, from http://www.fortune.com/fortune/smallbusiness/technology/articles/0,15114,1062892-1,00.html

Nisbett, R. E., Kaipeng, P., Incheol, C., & Norenzayan, A. (2001). Culture and systems of thought: Holistic versus analytical cognition. *Psychological Review, 108*, 291–310.

Pierce, T. (1996). *The international pictograms standard*. Cincinnati, OH: S. T. Publications.

Prabhu, G. V., Chen, B., Bubie, W., & Koch, C. (1997, 24–29 August). Internationalization and localization for cultural diversity. In Salvendy G. (Ed.), *Design of Computing Systems: Cognitive Considerations. Vol. 1: Proceedings of the 7th International Conference on Human-Computer Interaction* (HCI International '97) (pp. 149–152). Amsterdam: Elsevier.

Schwartz, S. H. (2004). Mapping and interpreting cultural differences around the world. In H. Vinken, J. Soeters, & P. Ester (Eds.), *Comparing cultures, dimensions of culture in a comparative perspective* (pp. 43–73). Leiden, Netherlands: Brill.

Singh, N. (2004). From cultural models to cultural categories: A framework for cultural analysis. *The Journal of American Academy of Business*, *5*(1/2), 1–8. [Vol 5, Nos. 1 & 2]

Singh, N., & Baack, D. W. (2004). website adaptation: A cross-cultural comparison of U. S. and Mexican websites. *Journal of Computer-Mediated Communication*, *9*(4), Retrieved 16 September 2006 from http://JCMC.Indiana.edu/vol9/issue4/singh_back.html

Spradley, J., & McCurdy, D. (1998). *The cultural experience: Ethnography in complex society.* Long Grove, IL: Waveland Press.

Stephanidis, C. (Ed.). (2000). *User interfaces for all.* New York: Lawrence Erlbaum Associates.

Trompenaars, F., & Turner, C. H. (1998). *Riding the waves of culture.* New York: McGraw-Hill.

Yeo, A. W. (2001). Global-software development lifecycle: An exploratory study. *Proceedings Computer-Human Interaction Conference 2001,* Seattle, WA, 104–111.

·19·

CONVERSATIONAL SPEECH INTERFACES AND TECHNOLOGIES

Jennifer Lai and Clare-Marie Karat
IBM T.J. *Watson Research Center*

Nicole Yankelovich
Sun Microsystems

Spoken Interface Lifecycle **383**
Selecting the Speech Technology **384**
How Does ASR Work? 384
How Does Text-to-Speech (TTS) Work? 384
Current Capabilities and Limitations of Speech-Synthesis Software 385
How Do Natural Language Processing (NLP) and Natural Language Understanding (NLU) Work? 385
Current Capabilities and Limitations of NLU Systems 386
Crafting the Spoken Interaction **386**
Dialog Styles 387
Directed dialogue (system initiated) 387
User-initiated 387
Conversational systems (mixed initiative) 387
Inherent Challenges of Speech 387
Prompt Design 387
Providing Help 388
Handling Errors 388
Rejection errors 389
Substitution errors 389
Insertion errors 389
Correction Strategies 389
Testing and Iterating **389**
Conclusion **390**
References **391**

A conversational speech interface for a computer emulates human-to-human interaction by calling on our inherent ability as humans to speak and listen. While human speech is a skill we acquire early and practice frequently, getting computers to map sounds to actions and to respond appropriately with either synthesized or recorded speech is a massive programming undertaking. Because we all speak a little differently from each other, and because the accuracy of the recognition is dependent on an audio signal that can be distorted by many factors, speech technology, like the other recognition technologies, lacks 100% accuracy. When designing a spoken interface, one must design to the strengths and weaknesses of the technology to optimize the overall user experience.

The goal for a spoken user interface is to emulate a conversation convincingly enough that the person interacting with the computer can use what he or she has learned in a lifetime of conversations. Successful communication occurs when the sender and receiver of the message achieve a shared understanding. Human-to-human conversations are characterized by turn taking, or shifts in initiative, as well as verbal and nonverbal feedback to indicate understanding. Herb Clark (1993) said, "Speaking and listening are two parts of a collective activity." Because language use is deeply ingrained in human behavior, successful speech interfaces should be based on an understanding of the different ways that people use language to communicate. Speech applications should adopt language conventions that help people know what they should say next and avoid conversational patterns that violate standards of polite, cooperative behavior.

Many excellent examples of speech user interfaces emulate effective conversational partners. In these systems, the computer "speaker" appears to remember contextual information and gives the impression of understanding what the user is saying. For example, several airlines use speech applications to handle telephone reservations or lost baggage tracking (Cohen, Giangola, & Balogh, 2004). These systems are a substantial improvement over previous interactive voice response (IVR) systems that relied solely on telephone keypad input. Instead of pressing "1" for this and "2" for that, users can speak natural language phrases such as "I'd like to travel from Boston to New York."

A crucial factor in determining the success of a spoken application is whether the use of speech technology presents a clear benefit. Speech is best used when it enables something that cannot otherwise be done, such as accessing electronic mail or an online calendar over the telephone when a computer is not available. In general, it is effective to use speech applications for situations when speech can enable a task to be done more efficiently, such as when a user's hands and eyes are busy doing another task. Likewise, speech input is useful when no keyboard is available for text entry, or when a user has a physical disability that limits the use of his or her hands, or if a user is just not comfortable typing. Speech output is particularly liberating for people with visual impairments. In addition, it provides a way of communicating information if the user is in a divided attention state such as driving, and it can be used to grab users' attention or embody a particular personality for a computer system or character.

While speech seems like it might be the ideal way to communicate with a computer anytime, there are situations when it is best not to use a speech user interface. For example, speech output is ineffective for delivering large amounts of information. Not only is it difficult for users to maintain the information in short-term memory, but people can read much faster than they can listen. Speech input can also be problematic when the speaker is not in a private environment, or when other voices in the background might interfere with the speech recognition.

The success of a spoken interaction with a computer depends on not only the task and the motivation of the user, but also on the physical device(s) being used. Speech input and output capabilities of devices greatly vary. For example, personal computers (PCs) provide good quality audio subsystems and speakers, and a variety of microphones that perform very well under quiet conditions can be used with them. The audio channels for other devices are not on par with PCs. While most personal digital assistants (PDAs) have started to offer speech input capabilities, the quality of their microphones and audio subsystems degrade the speech signal, resulting in poor recognition performance for many applications. In addition, PDAs have insufficient computing resources to enable large vocabulary speech recognition processing. For telephony applications, when the input device is either a cell phone or a landline, the speech recognition engines are usually deployed on large servers. The accuracy of telephony systems for a given task is normally lower than for a similar PC configuration with a headset microphone in a quiet environment. Background noise, signal degradation, poor cellular connection and the application of compression techniques can substantially reduce the performance of speech recognition systems. Nevertheless, careful design can compensate for many of these problems and lead to successful interaction with a telephone-based speech application.

Speech technology can be used for the following types of tasks:

- **Composition.** Composition tasks have as their primary goal the creation of a document such as word processing documents, e-mails, or instant-messaging text. Composition includes dictating the text and fixing any recognition errors.
- **Transcription.** Transcription is similar to composition in that a document is created from speech, but it differs by virtue of the fact that the primary user task is parallel to the creation of a document. Broadcast news, business meetings, and business calls are examples of situations where having a permanent, textual record of the speech is valuable. A digital (textual) record of the conversation is searchable, readable by the deaf, and supports advanced business intelligence applications such as data mining. These are all examples of transcription tasks.
- **Transaction.** The third type of interaction, and a major focus of this chapter, is one where users have as their goal the completion of one or more transactions, rather than the creation of a document or a permanent record of a conversation. Examples of transactional applications include financial account management such as trading stocks, e-commerce applications such as the purchase of computer equipment, searching for information on the Internet, and controlling the environment.
- **Collaboration.** The final type of conversational task is collaboration. Collaborative conversational applications are char-

acterized by tasks that result in human-to-human communication. One example of this is when speech recognition is used as an input modality to an instant messaging application.

Designing a speech user interface is similar to designing any other interface for human-computer interaction (HCI). A good design relies on applying principles of user-centered design, and many of these same principles and techniques can be used with speech interfaces. In this chapter, we discuss the lifecycle of a speech application from the starting point of understanding the requirements for speech, through knowing which technology to use, selecting a dialog style, and designing the prompts. All these steps culminate with an iterative testing process.

SPOKEN INTERFACE LIFECYCLE

Once a designer has clearly established that including speech technology for input, output, or both, is an appropriate design decision, the designer must begin to define how speech best fits into the accomplishment of the task. In order to do this, the designer must first have a clear understanding of the task. With the task well understood, the designer needs to overlay speech onto that model by listening to how people speak in the domain. Observing humans interacting and speaking to accomplish the task may result in additional refinements to the model. The next step is to select the appropriate speech technology, or combination of technologies, for the task, in the contexts of use and user group. The heart of the work for a speech interface designer is to craft the prompts that the system will "speak" and to prepare the system to recognize what the users will most likely say in response. Lastly, the designer should create a series of tests for the system, initially in the lab and ultimately out in the real world with users accomplishing real tasks. In between the various forms of testing, a designer needs to allocate time to refine the design and make changes based on what he or she learns from the results of the testing. In this chapter, we will cover certain steps in greater detail than others. These steps include (a) listening to users speak when accomplishing the task, (b) selecting the technology, and (c) designing the system prompts.

Thus, the first step in the life cycle of a conversational interface is to model the task that users will be accomplishing. This requires the designer to sketch out the logical steps that make up the task, along with the pieces of information that must be exchanged in order to complete each step. With a speech application, each interaction between the system and a user is often referred to as a "turn." Thus, the *task model* tries to capture the expected number of turns, as well as the vocabulary that is required to support the information exchange during those turns. It is important for the designer to consider alternate flows, since not all users will necessarily approach the task in the same way. For example, when making a flight reservation, some users will select a flight based exclusively on time schedules, while others will opt for airline loyalty and price points, accepting any flight time that meets their criteria for loyalty and cost.

The model must operate within the set of constraints for the application. Constraints usually include the business goals for creating the speech application, as well as the requirements resulting from the environment in which users may find themselves (e.g., noisy environment) and the users themselves (e.g., the majority of users will be over 65 years old and uncomfortable with technology). For example, if a business goal is to cross-sell related items when a customer makes a purchase by calling into a speech-enabled call center application, the designer will want to incorporate a turn that involves the system suggesting related items that are on sale that day.

SELECTING THE SPEECH TECHNOLOGY

While good interaction design in a speech application can compensate for some shortcomings in speech technology, if a certain baseline level of accuracy is not achieved, the application probably will not succeed. Accuracy depends on the choice of the underlying speech technology, and making the best match between the technology, the task, the users, and the context of use. Automatic Speech Recognition (ASR) can have explicitly defined, rule-based grammars, or use statistical grammars such as language models. Usually, a transactional system uses explicitly defined grammars, while dictation systems or natural language understanding systems use statistical models. The designer will also have to decide whether to use synthetic speech in the system or not. Synthetic speech, also known as text-to-speech (TTS), is speech produced by a computer. Given that today's synthesizers still do not sound entirely natural, the choice of whether to use synthesized output, recorded output, or no speech output can be a difficult one. The next section discusses the various technologies in more detail.

How Does ASR Work?

ASR systems work by analyzing the acoustic signal received through a microphone connected to the computer (see Figure 19.1). The user speaks and the microphone captures the acoustic signal as digital data, which is then analyzed by an acoustic model and a language model. The different speech recognition systems on the market differ in the building blocks (e.g., phonemes) that they employ to analyze the acoustic signal data. The analysis employs algorithms based on Hidden Markov Models, a type of algorithm that uses stochastic modeling to decode a sequence of symbols, to complete the computations (Rabiner, 1989; Roe & Wilpon, 1993). After the acoustic analysis

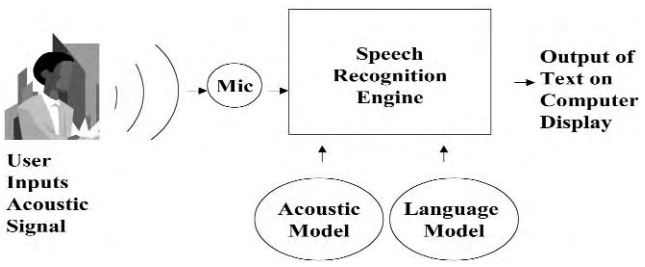


FIGURE 19.1. Overview of human-computer interaction model for speech recognition.

is complete, the system analyzes the resulting strings of building-block data using a language model that contains a base vocabulary and any specific domain topics (e.g., computer, medical) that may have been added. When the analysis is completed, the text appears on the computer screen.

Speech recognition systems require computers with an approximate minimum of 200MHz processor, 32MB of RAM, 300 MB of available hard disk, and a 16-bit sound card with a microphone input jack and good recording. These requirements enable local decoding of the recognized speech. It is possible to have a system, such as a small, pervasive device, capture a user's speech and then decode it on a remote server (e.g., Price & Sears, 2005) and return the decoded text to the user, albeit with a short time delay. Multiple users can work with one installation of a speech recognition system. In some systems, users create personal voice models and log on with their individual user names, and the system uses their personalized speech files for the recognition processing. Each user can also create several different user voice models in order to achieve the best recognition rates in environments with different levels of background noise (e.g., home, office, and mobile work locations). For results on the effectiveness of creating and using different user voice models, see Price, Ling, Feng, Goldman, Sears, & Jacko (2005).

Karat, Lai, Danis, and Wolf (1999) reported ASR system accuracy rates in the mid-90s for "in vocabulary" words (e.g., words in the 20,000-word vocabulary included with the software) in people's initial use of continuous speech recognition software. Error rates of 2–5% are common with more frequent use. Karat, Horn, Halverson, and J. Karat (1999) tested three commercially available speech recognition systems in 1998 with users in initial and extended use and found initial use data of 13.6 corrected words per minute with an average of one incorrect, missing, or extra word for every 20 words and one formatting error every 75 words. Improved ASR system accuracy and higher user productivity were measured when users employed more recent versions of the speech recognition software (Sears, Karat, Oseitutu, Kaimullah, & Feng, 2000; Feng, Sears, & Karat, 2006), since users are able to correct errors more easily and quickly. The product also provided a better quality microphone to reduce the number of recognition errors that occurred in the first place.

In general, there are three types of errors that occur in ASR systems (Halverson, Horn, C. Karat, & J. Karat, 1999). The first is that users can make direct errors where they misspeak, stutter, or press the wrong key. Second, users can make errors of intent, where they decide to restate something. The third type of error is an indirect error, where the speech recognition system mis-recognizes what the user says. The indirect errors are difficult to detect during proofreading. All three types of errors can lead to cascading errors where, in the process of correcting one error, others occur.

Most telephony systems are speaker-independent (e.g., no personalized training of the voice models required) speech-recognition systems. They are also usually server based and must handle the signal degradation that occurs across the telephone lines. Telephony systems can be created from a combination of speech recognition and natural language processing (NLP) technologies. In telephony systems, a dialog manager component works with the speech recognition software to handle the course of the conversation with the user. The system provides feedback to the user through the dialog manager using recordings of either human voice or text-to-speech. Telephony systems have capabilities such as "barge in" and "talk ahead" that enable the user to redirect the action of the system and complete multiple requests before being prompted for additional information necessary to complete the task. Conversational telephony systems include interactive voice response (IVR) systems and new systems built using voice XML (see section on NLP for description of the technology).

How Does Text-to-Speech (TTS) Work?

Text-to-speech (TTS) synthesis enables computers or other electronic systems such as telephones to output simulated human speech. Synthetic speech is based on the fields of (a) text analysis, (b) phonetics, (c) phonology, (d) syntax, (e) acoustic phonetics, and (f) signal processing. The quality and effectiveness of speech synthesis is based on a hierarchy. The base level of achievement is to produce speech synthesis that is intelligible by human beings. The second level is to produce speech synthesis that simulates the natural qualities of human speech. The third level of speech synthesis is to produce synthesized speech that is personalized to the person it is representing—that is, the intonation of the particular person's speech is represented. The fourth and highest level of achievement in synthesized speech is to produce speech based on a person's own voice recordings, so that the speech sounds just like the actual person being represented. Currently, speech synthesis technology has achieved the base level of quality and effectiveness, and concatenated synthesis can simulate the natural quality of human speech, although at great expense. The third and fourth levels of speech synthesis technology are the focus of research in laboratories around the world.

Two types of speech synthesis are commercially available today: (a) concatenated synthesis and (b) formant synthesis, which is the most prevalent type. Concatenated synthesis employs computers to assemble recorded voice sounds into speech output. It sounds natural, but can be prohibitively expensive for many applications, as it requires large disk storage space for the units of recorded speech and significant computational power to assemble the speech units on demand. Concatenated synthesizers rely on databases of diphones and demi-syllables to create the natural-sounding synthesized speech. Diphones are the transitions between phonemes. Demi-syllables are the half-syllables recorded from the beginning of a sound to the center point, or from the center point to the end of a sound (Weinschenk & Barker, 2000). After the voice units are recorded, then the database of units is coded for changes in frequency, pitch, and prosody (intonation and duration). The coding process enables the database of voice units to be as efficient as possible.

Formant synthesis is a rule-based process that creates machine-generated speech (see Figure 19.2). A set of phonological rules is applied to an audio waveform that simulates human speech. Formant synthesis involves two complex steps: the first covers the conversion of the input text into a phonetic representation, while the second encompasses the production of sound based on that phonetic representation. In the first step, the text is input from a database or file and is normalized so that any symbols or abbreviations are resolved as full alphabetic words. To convert the words into phonemes, a pronunciation dictionary is used for most words and a set of letter-to-sound

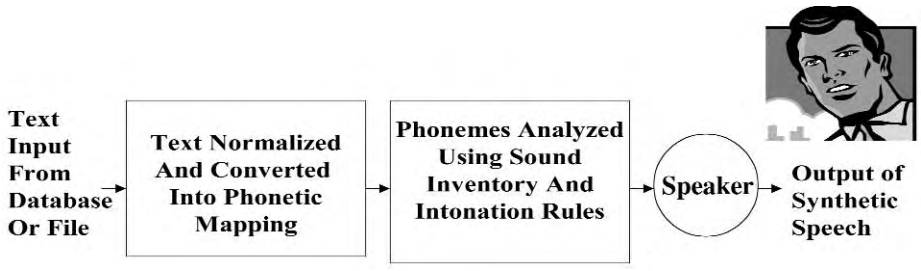


FIGURE 19.2. Model of text to speech synthesis.

rules is used for word exceptions not available in the dictionary. In the second step, the phonemes are analyzed using a sound inventory and intonation rules about pitch and duration. The speech synthesis is the resulting output heard by users through a speaker or headphone.

The quality of the synthetic speech should be evaluated along the lines of its acceptability, naturalness, and intelligibility. It is important to ask users to evaluate different speech against each other, because these qualities are always relative (Francis & Nusbaum, 1999). Although these subjective differences in opinion will probably always exist, a study by Lai, Wood, and Considine (2000) showed that there were no significant differences in comprehension levels for longer messages (e.g., with a word length ranging from 100 to 500 words) among five major commercial text-to-speech engines.

Current Capabilities and Limitations of Speech-Synthesis Software

Formant synthesis produces speech that is highly intelligible, but sounds unnatural; however, it has the power to inexpensively produce nearly unlimited speech, from a resource point of view. The limitation of using formant synthesis is the complexity of the required linguistic rules to produce accurate speech output. Utilizing domain-specific information and assumptions produces a substantial improvement in the synthesizer's prosody. (*Prosody* refers to speech elements such as intonation, duration, pitch, rate, loudness, or rhythm.) Users will be able to comprehend the speech at a higher rate and will perceive the voice to be more natural.

Some applications of concatenated synthesis attempt to reduce costs by basing the voice recordings on whole words. Interactive voice systems (IVR) and voice mail systems that use concatenated speech synthesis based on voice recordings of whole words sound unnatural and unevenly paced, and this makes the synthetic voice hard to understand or remember. An application of concatenated synthesis needs to be done correctly or not at all.

How Do Natural Language Processing (NLP) and Natural Language Understanding (NLU) Work?

"Natural language processing (NLP)" refers to a wide range of processing techniques aimed at extracting, representing, responding to, and ultimately understanding the semantics of text. Natural Language Understanding (NLU) is an area of NLP focused on the understanding of natural language text. It is the process of analyzing text and taking some action based on the meaning of the text. We include any technology that allows a user to communicate with a system using a language that is not rigidly structured (e.g, a "formal" language). The focus of this chapter is on systems in which communication between the user and the system has constructs similar in grammar and dialog to the language of everyday human-to-human communication.

As a technology, NLU is independent from speech recognition, although the combination of the two yields a powerful human-computer interaction (HCI) paradigm (see Figure 19.3). When combined with NLU, speech recognition transcribes an acoustic signal into text, which is then interpreted by an understanding component to extract meaning. In a conversational system, a dialog manager will then determine the appropriate response to give the user. Communication with the user can take place via a variety of modalities, including (a) speech input and output, (b) text input and output, (c) handwriting, or (d) some combination of these modalities. Figure 19.3 shows a block diagram of a prototypical multimodal conversational system that allows speech and keyboard natural language input, and speech and GUI text output.

NLU has been an active area of research for many decades. The promise of NLU lies in the "naturalness" of the interaction. Since humans have deep expertise in interacting with one another through language, the idea that leveraging a user's ability to interact using language will result in systems with greater usability has been an implicit and explicit hypothesis in a wide array of research studies and technology development efforts. Designing systems that use natural interaction techniques means the user is freed from learning the formal language of a system. Thus, instead of using formalisms (e.g., UNIX commands), scripting languages, or graphical menus and buttons, users can engage in a dialog with the system. Less user training, more rapid development of expertise and better error recovery are all promising aspects of systems that use NLU.

Dialog can be described as a series of related conversational interactions. It adds the richness of context and the knowledge of multiple interactions over time to the user interface, transforming natural language interfaces into conversational interfaces. Such interactions require the system to maintain a history of the interaction as well as the state of the interaction at all times (Chai, 2001). Two important components, which must be represented in any dialog system, are user goals and the current context of the interaction. The history of the interaction must be evaluated against the user goal, with prompts to the user designed to acquire the necessary information required to satisfy a goal. Dialog

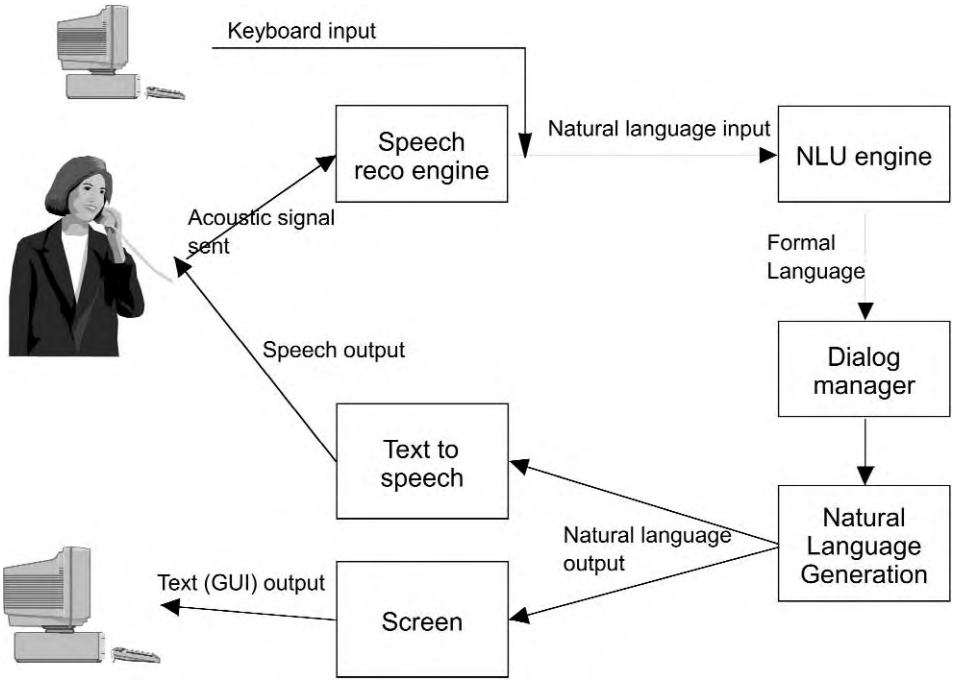


FIGURE 19.3. Diagram of a prototypical multimodal conversational system.

technology can be embedded in an application to enable either user-initiated, system-initiated, or mixed-initiative conversations (see descriptions of these different dialog styles following). The conversations are guided by dialog management technology.

Current Capabilities and Limitations of NLU Systems

Most NLU applications will easily run on the desktop and notebook computers commonly available in the year 2007. In addition, research prototype systems that move NLU functionality onto handheld systems are beginning to appear. Cars that accept natural voice commands to control a limited range of features are now available. NLU on the web is becoming more common (e.g., www.askjeeves.com, www.neuromedia.com), and illustrates client-server architecture for NLU. As computing power increases and digital technology pervades our everyday life, we can expect to see a dramatic increase in the use of NLU in new environments, supporting more everyday tasks.

Capturing and representing knowledge of a domain is a complex and labor-intensive process. As the scope of the application domain increases, it becomes increasingly difficult to build and maintain NLU applications. As a result, all successful examples of rich NLU interfaces have relatively narrow application domains. This has direct impact on the definition of user profiles for any given application. Simply stated, the narrower the conceptual, functional, syntactic, and lexical domains of the target user population, the greater the chance of building an NLU application that satisfies its users. There is currently no way to quantify and measure these characteristics, and determining whether a particular application represents a tractable problem in NLU is largely an issue of experience and instinct on the part of designers and engineers. An NLU engine and speech input and output technologies are the building blocks for a conversational application. Now, let's turn our attention to the design of the application itself.

CRAFTING THE SPOKEN INTERACTION

The first step in designing a conversational interface is to educate yourself about the ways people speak in the domain of the task. The best approach is to find users who are doing an activity that is as close as possible to the target task. Keep in mind that the vocabulary and sentence structure used in graphical computer applications or in printed material may be quite different from the way people actually speak about a task. For example, imagine that you want to provide a speech interface to a calendar. Printed calendars and online calendars typically show day, week, and month views of calendar appointments. They always show days of the week and numbered dates. As soon as you listen to two people talk about appointments and scheduling, however, you learn that they only use numbered dates occasionally. The more common way to discuss dates with other people is to use relative descriptions: "What do you have a week from Monday? Are you busy this Thursday? What about the next day?" The concept of relative dates is completely absent from paper-based and online calendars; however, users will expect to be able to use them when speaking.

With a little creativity, you can find or create situations in which two people can talk about the target task. Although you might want to plan out the interactions a little bit, care must be taken not to use scripts or memorized phrases. The idea with this type of listening is to allow people's natural social instincts to drive the

dialog. If possible, record the natural dialogs. They will serve as the basis for the speech interface design. Listening in this way is called a "natural dialogue study" (Yankelovich, in press). Using natural dialogue studies, you can learn (a) about concepts used when talking about the task that are absent in other media, (b) common vocabulary used when talking about the task, (c) tone of voice that is considered polite in the context of the task, (d) sentence structure typical for the domain patterns of interaction, and (e) methods people use to give one another feedback.

Dialog Styles

Once you have an understanding of your task and the way people speak in this domain, you will need to decide if your application would benefit most from a directed, user-initiated, or conversational dialogue style.

Directed dialogue (system initiated). This is the most commonly used style of interaction in speech-based telephony systems on the market today. With a directed dialogue, the user is instructed or "directed" what to say at each prompt (Kamm, 1994). Systems that use this style can often be recognized by the use of the word "say" in the prompts. "Welcome to ABC Bank. You can check an account balance, transfer funds, or pay a bill. Say, 'Balance,' 'Transfer,' or 'Pay.'" The reason most systems use this dialogue style is to increase the accuracy of the speech recognition. By telling the user what to say, the chances of the user speaking an out-of-vocabulary utterance are much lower. This style is sometimes referred to as "system initiated" because the user plays a more passive role, waiting for the system to prompt the user for what is needed.

User-initiated. Unlike a directed dialogue, which works well for novice users, user-initiated systems are intended for use by expert users. With this type of interaction style, the system is passive and the user is not prompted for specific information. For example,

Computer: How can I help you?

User: I would like to buy 100 shares of Amazon.com at the market price with funds from my current account number 345198.

These types of dialogues do not work well with new users who do not know what to say, and tend to have lower recognition rates.

Conversational systems (mixed initiative). The type of interaction that is easiest for users and feels the most natural is a mixed initiative system. In this style of interaction, the system and the user take turns initiating the communication depending on the flow of the conversation and the status of the task. With a mixed initiative system, the prompts can fluctuate between open-ended prompts and more directed prompts, depending on the context. Here's a portion of dialogue:

Computer: What would you like to do? (*Open-ended*)

User: What is the price of Fidelity Magellan? (*User-initiated*)

Computer: Price of Fidelity Magellan is $36.05 per share. What else?

User: I want to buy 500 shares. (*Which shares to buy is implied in the previous interaction.*)

Computer: Confirming purchase of 500 shares of Fidelity Magellan fund, please say, "Yes" or "No." (*Directed*)

User: No, skip the purchase. I'd like to transfer $1,000 from my index fund (*User-initiated*).

Computer: Purchase canceled. To which fund would you like to transfer? *(Prompt for missing information*)

In this example, we see the system functioning as a cooperative partner in the conversation.

Inherent Challenges of Speech

Designing conversational systems poses substantial challenges. Understanding these challenges and assessing the various trade-offs that must be made during the design process will help to produce the most effective interface.

By its nature, speech is *transient*. Once you hear it or say it, it's gone. By contrast, graphics are *persistent*. A graphical interface typically stays on the screen until the user performs some action. Listening to speech taxes users' short-term memory. Because speech is transient, users can remember only a limited number of items in a list, and they may forget important information provided at the beginning of a long sentence. Users' limited ability to remember transient information has substantial implications for the speech interface design. In general, transience means that speech is not a good medium for delivering large amounts of information. The transient nature of speech can also provide benefits. Because people can look and listen at the same time, speech is ideal for grabbing attention or for providing an alternate mechanism for feedback. Imagine receiving a notification about the arrival of an e-mail message while working on a spreadsheet. Speech might give the user the opportunity to ask for the sender or the subject of the message. The information can be delivered without forcing the user to switch contexts.

Speech is also *invisible*. The lack of visibility makes it challenging to communicate the functional boundaries of an application to the user (Yankelovich, 1996). In a graphical application, menus and other screen elements make most or all of the functionality of an application visible to a user. By contrast, in a speech application, it is much more difficult to indicate to the user what actions they may perform, and what words and phrases they must say to perform those actions.

Prompt Design

The problems of transience and invisibility can often be mitigated by well-designed prompts. Many factors must be considered when designing prompts, but the most important is assessing the trade-off between flexibility and performance. The more you

constrain what the user can say to an application, the less likely they are to encounter recognition errors. On the other hand, allowing users to speak information flexibly can often speed the interaction (if recognition succeeds), feel more natural, and avoid forcing users to memorize commands. Here are some tips for creating useful prompts.

Use *explicit prompts* when the user input must be tightly constrained. For example, after recording a message, the prompt might be "Say, 'Cancel,' 'Send,' or 'Review.'" This sort of prompt directs the user to say just one of those three keywords. Even conversational systems should fall back on this type of prompt when it is critical that the input be correctly recognized, as with transactions that involve transferring money or booking travel. "Are you sure you want to book this flight from JFK to LAX? Please say, 'Yes' or 'No.'"

Use *implicit prompts* when the application is able to accept input that is more flexible. These prompts rely on conversational conventions to constrain the user input. For example, if the user says, "Send mail to Bill," and "Bill" is ambiguous, the system prompt might be "Did you mean Bill Smith or Bill Jones?" Users are likely to respond with input such as "Smith" or "I meant Bill Jones." While possible, conversational convention makes it less likely that they would say, "Bill Jones is the one I want."

Using *variable prompts* is a good way to try to simulate a human-human conversation. Given a certain condition or state of the system (e.g., the ready state, or a system response to silence), it is preferable not to play the exact same system prompt every time. Subtle variations in the wording impart a much more natural feel to the interaction. Note the following possibilities for the ready state in a natural language understanding system: "What now?", "I'm ready to help," "What's next?"

Another interaction that we can model on human speech is the use of *tapered prompts*. Tapering can be accomplished in one of two ways. If an application is presenting a set of data such as current quotes for a stock portfolio, drop out unnecessary words once a pattern is established. For example, "As of 15 minutes ago, Acme Industries was trading at 45 up 1/2, BioStartup was at 83 up 1/2, and TechGiant was at 106 down 1/4." Tapering can also happen over time—that is, if you need to tell the user the same information more than once, make it shorter each time the function is used.

One way to speed interaction for expert users and provide help for less experienced users is to use *incremental prompts*. This technique involves starting with a short prompt. If the user does not respond within a time-out period, the application prompts again with instructions that are more detailed. For example, the initial prompt might be, "Which service?" If the user says nothing, then the prompt could be expanded to "Say 'Banking,' 'Address book,' or 'Yellow pages.'" Incremental prompts have been used successfully in a number of systems, but they suffer from several problems. A first-time user is just as likely to say something to the first prompt as they are to say nothing. This often results in a recognition error. Another common pitfall with incremental prompts is that they tend to cause collision errors where both the system and the user speak at the same time. It is common for timing problems to happen repeatedly once started, increasing the user's frustration along the way.

Providing Help

There are two basic types of help. In the first case, the user initiates by requesting help. Common queries for help in speech systems are, "Help", "What can I say?", or "What are my choices?" It is always a good idea to support several variations of this request in the grammar. In the second type of help, the system detects that the user is in trouble and automatically presents help. Although not all forms of user problems are detectable by the system, one can presume if a series of errors or long silences occur that the user probably needs some assistance.

Help, when given, needs to be specific to the user's current task. Few things are as frustrating as being stuck and asking for help only to find that the information presented, while long and detailed, has no bearing on the current situation. A good approach for help is to provide the user with an example of a successful utterance just as the user starts the downward spiral of repeated errors. If several errors have occurred in a row, the system should present the user with sample phrasing that would be expected to work, based on the context that he is in. For example, a user could say, "I want to sell 300 shares of Ultrasoft." If the system cannot determine the exact context, another option is to list the functions that are available at the current branch and have the user point the system in the right direction. Here's an example:

Computer:	Let's start over. You can forward, reply, delete, or create a new message. What do you want to do?
User:	I want to forward a message.
Computer:	To forward a message, say the first and last name of another subscriber. For example, say, "Forward this message to John Smith."
User:	Forward this message to Alishia Grant.

If the system returns a recognition error now, it is easy to establish that the name is at fault because we already know the function. This dialogue might continue:

Computer:	Oh dear. I didn't catch that name. Just say the first name.
User:	Alishia.
Computer:	I think you said, "Alice." Is that correct?
User:	No, that is not correct.
Computer:	Thanks for your patience with this. Please spell the first name.
User:	A-L-I-S-H-I-A.

Now the system can repeat a similar process for the last name, or can look up all subscribers with a first name of Alishia and offer the user a set of possible last names from which to choose.

Handling Errors

How a system handles recognition errors can dramatically affect the quality of a user's experience. If either the application or the

user detects an error, an effective speech user interface should provide one or more mechanisms for correcting the error. Although this seems obvious, correcting a speech input error is not always easy! If the user speaks a word or phrase again, the same error could reoccur, depending on the cause of the error (Yankelovich, Levow, & Marx, 1995).

Recognition errors can be divided into three categories: (a) rejection, (b) substitution, and (c) insertion (Schmandt, 1994; Ballentine & Morgan, 1999). A rejection error occurs when the recognizer has no hypothesis about what the user said. A substitution error involves the recognizer mistaking the user's utterance for a different valid utterance, as when "send a message" is interpreted as "seventh message." With an insertion error, the recognizer either interprets noise as a valid utterance, or decodes multiple words when only one was spoken. This can be caused by other people talking nearby or by the user inadvertently tapping the telephone or microphone.

Rejection errors. In handling rejection errors, designers want to avoid the "brick wall" effect, when every rejection is met with the same "I didn't understand" response. Users get frustrated very quickly when faced with repetitive error messages. Instead, give *progressive assistance*: a short error message the first couple of times, and if errors persist, offer more detailed assistance. For example, here is one progression of error messages that a user might encounter: "Sorry?", "What did you say?", "Sorry. Please rephrase.", "I didn't understand. Speak clearly, but do not overemphasize.", "Still no luck. Wait for the prompt tone before speaking." Progressive assistance does more than bring the error to the user's attention; the user is guided toward speaking a valid utterance with successively more informative error messages that consider possible causes of the error.

Substitution errors. Although rejection errors are frustrating, substitution errors can be damaging. If the user asks a weather application for "Kuai," but the recognizer hears "Goodbye" and hangs up, the interaction could be completely terminated. In situations like this, the system should explicitly verify that the user's utterance was correctly understood. Verification should be commensurate with the cost of the action that would be effected by the recognized utterance. Reading the wrong stock quote or calendar entry will make the user wait a few seconds, but hanging up or sending a confidential message to the wrong person by mistake could have serious consequences.

Insertion errors. Spurious recognition typically occurs because of background noise. The illusory utterance will either be rejected or mistaken for an actual command; in either case, the previous methods can be applied. The real challenge is to prevent insertion errors. One option is to provide users with a keypad command to turn off the speech recognizer to talk to someone, sneeze, or simply gather their thoughts. Pressing the keypad command again can restart the recognizer with a simple prompt, such as "What now?" to indicate that the recognizer is listening again.

Whatever the type of error, a general technique for avoiding errors in the first place is to filter recognition results for unlikely user input. For example, a scheduling application might assume that an error has occurred if the user appears to want to schedule a meeting for three o'clock in the morning.

Correction Strategies

If errors do occur, it is important to provide a means for the user to correct the error (assuming they notice it). Flexible correction mechanisms that allow a user to correct a portion of the input are helpful. For example, if the user asks for a weather forecast for Boston for Tuesday, the system might respond, "Thursday's weather for Boston is . . . ." A flexible correction mechanism would allow the user to just correct the day: "No, I said Tuesday."

When possible, using an alternate form of input, such as the telephone keypad, can alleviate the user's frustration. If the user is at a prompt where three choices are available and the user has encountered several rejection errors, the user could be instructed: "Press any key when you hear the option you want." The telephone keypad also works well when the requested input is numeric (e.g., telephone numbers, account or social security numbers). Getting users to type alphabetic text using a telephone keypad is not a good idea; speech systems are usually recommended to avoid this type of input in the first place.

Another strategy is to have the system take its best guess at the requested function. This is a good tactic to take when the number of functions enabled at a particular branch in the dialogue is too large to list for the user. A reasonable prompt is, "I think you are trying to create a message, is that correct?" If the user answers in the affirmative, the conversation moves forward and the system can present the user with a sample valid utterance for that function. If the best guess is wrong, however, it is not a good idea to keep iterating through the N-best choices because this only leads to user frustration. If the response to the best guess is negative, a better solution is to reprompt with a restricted set of choices. Be sure to eliminate the choice that is definitely wrong (e.g., creating a message in the previous example). The goal is to move away from a very general prompt such as, "I'm sorry, I do not understand. Please try again" towards a directed prompt that will increase the likelihood of success. A series of errors in a row is a clear indication that simply having the user repeat the utterance, or rephrase it, is not working.

The best-guess tactic can be combined with another correction strategy, switching to more constrained grammar, to increase its likelihood of success. For example, in the prior prompt, a directive of what utterances are available to the user can be added: "I think you are trying to create a message. Is that correct? Please say, 'Yes' or 'No.'"

TESTING AND ITERATING

Once the preliminary application design is complete, a Wizard of Oz study can help test and refine the interface. The speech data collected from this study can be used to help refine the grammar (Rudnicky, 1995) or, in the case of a NLU system, to statistically train the engine on potential utterances. In these studies, a human wizard—often using software tools—simulates

the speech interface. Major usability problems are often uncovered with these types of simulations.

A Wizard of Oz study usually involves bringing participants into a lab and telling them they will be interacting with a computer. If it is a telephony application, they can be asked to call a telephone number, at which point a human "wizard" picks up the phone and manipulates software so that recordings of synthesized speech are played to the participant. As the participant makes requests to the computer, the wizard carries out the operations and has the computer speak the responses. Often, none of the participants suspect that they are not interacting with a real speech system (Dahlback, 1993). Because computer tools are usually necessary to carry out a convincing simulation, Wizard of Oz studies are more time-consuming and complicated to run than natural dialogue studies. If a prototype of the final application can be built quickly, it may be more cost-effective to move directly to a usability study.

With speech applications, usability studies are particularly important for uncovering problems because of recognition errors, which are difficult to simulate effectively in a Wizard of Oz study, but are a leading cause of usability problems. The effectiveness of an application's error recovery functionality must be tested in the environments in which real users will use the application. Conducting usability tests of speech applications can be a bit tricky. Two standard techniques used in tests of graphical applications—(a) facilitated discussions and (b) speak-aloud protocols—cannot be used effectively with speech applications. It is best to have study participants work in isolation, speaking only into a telephone or microphone. A tester should not intervene unless the participant becomes completely stuck. A follow-up interview can be used to collect the participant's comments and reactions. Ultimately, the system needs to be tested with real users, accomplishing real tasks in a realistic environment. This type of testing will provide data that will allow the designer to modify the grammar and prompts as necessary.

CONCLUSION

The design and development life cycle for an effective speech application differs from a traditional GUI application. An effective speech application is one that uses speech to enhance a user's performance of a task or enable an activity that cannot be done without it; however, the design process does share common elements with the design of a successful GUI application, such as the need to first understand the task that the users are trying to accomplish. In the case of speech, modeling the task can usually be accomplished by listening to how humans accomplish the task today. Sometimes this involves listening to the interactions between a call center employee and a caller; sometimes it involves constructing a situation between two humans and asking them to converse to accomplish the task. Observing users during the task-modeling phase helps a designer to understand who the users are and what their goals are. Listening carefully to users while conducting natural dialogue studies shows how they speak in the context of the task. A natural dialogue study also ensures that prompts and feedback follow the conversational conventions that users expect in a cooperative interaction.

Once the task is modeled and well understood, and the business goals have been defined, the designer/developer must select the appropriate technology for the task, users and context of use. Conversational interfaces that use NLU technologies capitalize on human expertise in interacting with each other through language. Not all tasks will be able to use NLU, given the resource requirements and the need for a constrained domain; however, even speech applications that are built with grammars can carry many of the usability advantages of NLU and have a conversational feel to them, if the prompts are carefully crafted. Successful dialogs will move between user-initiated, system-initiated, and mixed-initiative styles depending on the state of the task and the history of the interaction.

Another key to the success of the design of an effective speech user interface is to follow an iterative process. Not even the most experienced designer will craft a perfect dialogue in the first iteration. Once an application is designed, Wizard of Oz and usability studies provide opportunities to test interaction techniques and refine application behavior based on feedback from prototypical users. To ensure that the system will be used over time, the designer must (a) focus on users and (b) modify the design as new data is collected.

Finally, testing the design with target users ensures that the prompts are clear, that feedback is appropriate, and that errors are caught and corrected. In addition, as part of the testing, verifying that the design accomplishes the business goals that were set out to be achieved helps the application gain approval from the sponsors. If problems are uncovered during testing, the design should be revised and tested again. By focusing on users and iterating on the design, one can produce an effective, polished speech interface design.

References

Ballentine, B., & Morgan, D. (1999). *How to build a speech recognition application: A style guide for telephony dialogues*. San Ramon, CA: Enterprise Integration Group, Inc.

Chai, J. (2001). *Natural language sales assistant—a Web-based dialog system for online sales*. To be presented at the 13th Conference on Innovative Applications of Artificial Intelligence.

Clark, H. (1993). *Arenas of language use*. Chicago: University of Chicago Press.

Cohen, M., Giangola, J., & Balogh, J. (2004). *Voice user interface design.* Boston: Addison-Wesley.

Dahlback, N., Jonsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies—why and how. *Proceedings of the Third Conference on Applied Natural Language Processing,* Trento, Italy, April 1993.

Feng, J., Sears, A., & Karat, C. (2006). A longitudinal evaluation of hands-free speech-based navigation during dictation. *International Journal of Human Computer Studies, 64*, 6, 553–569.

Francis, A., & Nusbaum, H. (1999). Evaluating the quality of synthetic speech. In D. Gardner-Bonneau (Ed.), *Human factors and voice interactive systems* (pp. 25–47). Boston: Kluwer Academic Publishers.

Halverson, C. A., Horn, D. A., Karat, C., & Karat, J. (1999). The beauty of errors: Patterns of error correction in desktop speech systems. In M. A. Sasse & C. Johnson (Eds.), *Human-computer interaction—INTERACT '99* (pp. 133–140). Edinburgh: IOS Press.

Kamm, C. (1994). User interfaces for voice applications. In D. B. Roe & J. G. Wilpon (Eds.), *Voice communication between humans and machines* (pp. 422–442). Washington, DC: National Academy Press.

Karat, C., Horn, D., Halverson, C., & Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. In M. Altom & M. Williams (Eds.), *Human Factors in Computing Systems—CHI 99 Conference Proceedings* (pp. 568–576). New York: ACM.

Karat, J., Horn, D., Halverson, C., & Karat, C. (2000). Overcoming unusability: Developing efficient strategies in speech recognition systems. In Karat, C.-M., Lund, A., Coutaz, J., & Karat, J. (Eds), *Human Factors in Computing Systems—CHI 2000 Conference Proceedings* New York: ACM.

Karat, J., Lai, J., Danis, C., & Wolf, C. (1999). Speech user interface evolution. In D. Gardner-Bonneau (Ed.), *Human factors and voice interactive systems* (pp. 1–35). Boston: Kluwer.

Lai, J., Wood, D., & Considine, M. (2000). The effect of task conditions on the comprehensibility of speech. *Human Factors in Computing Systems—CHI 2000 Conference Proceedings.* The Hague, Netherlands.

Price, K., & Sears, A. (2005). Speech-based text entry for mobile handheld devices: An analysis of efficacy and error correction techniques for server-based solutions. *International Journal of Human-Computer Interaction, 19*(3), 279–304

Price, K. J., Lin, M., Feng, J., Goldman, R., Sears, A., & Jacko, J. A. (in press). Motion does matter: An examination of speech-based text entry on the move. *Universal Access in the Information Society.*

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE, 77*, 257–286.

Roe, D. B., & Wilpon, J. G. (1993). Wither speech recognition: The next 25 years. *IEEE Communications Magazine, 11*, 54–62.

Rudnicky, A. (1995). The design of spoken language interfaces. In A. Syrdal, R. Bennett, & S. Greenspan (Eds.), *Applied speech technology* (pp. 18–39). CRC Press.

Schmandt, C. (1994). *Voice communication with computers: Conversational systems*. New York: Van Nostrand Reinhold.

Sears, A., Karat, C., Oseitutu, K., Kaimullah, A., & Feng, J. (2000). Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society, 1*, 5–25.

Weinshenk, S., & Barker, D. T. (2000). *Designing effective speech user interfaces*. New York: Wiley.

Yankelovich, N., Levow, G. A., & Marx, M. (1995, May 7–11). Designing SpeechActs: Issues in speech user interfaces. *Proceedings of the '95 Conference on Human Factors in Computing Systems*, Denver, CO.

Yankelovich, N. (1996, November/December). How do users know what to say? *ACM Interactions*, *3* (pp. 32–43).

Yankelovich, N. (in press). Using natural dialogs as the basis for speech interface design. In D. Gardner-Bonneau & H. E. Blanchard (Eds.), *Human factors and voice interactive systems* (2nd ed.). Springer.

·20·

MULTIMEDIA USER INTERFACE DESIGN

Alistair Sutcliffe
Manchester Business School
University of Manchester

Definitions and Terminology **394**
Cognitive Background **395**
Perception and Comprehension 395
Selective Attention 395
Emotion and Arousal 396
Learning and Memorization 396
Design Process **397**
Users, Requirements, and Domains 398
Information Architecture 398
Media Selection and Combination 400
Media selection 401
Engagement and attractiveness 403
Aesthetic design 403
Image and identity 404
Affective effects 404
Interaction and Navigation 405
Metaphors and interaction design 405
Navigation 406
Design for Attention 407
Still image media 409
Moving image media 409
Linguistic media (text and speech) 409
Conclusions **410**
References **410**

Design of multimedia interfaces currently leaves a lot to be desired. As with many emerging technologies, it is the fascination with new devices, functions, and forms of interaction that has motivated design rather than ease of use, or even utility of practical applications. Poor usability limits the effectiveness of multimedia products which might look good, but do not deliver effective use (Scaife, Rogers, Aldrich, & Davies, 1997). The multimedia market has progressed beyond the initial hype, and customers are looking for well-designed, effective, and mature products.

The distinguishing characteristics of multimedia are information-intensive applications that have complex design spaces for presenting information to people. Designers, therefore, must start by modeling information requirements. This chapter describes a design process that starts with an information analysis and then progresses to deal with issues of media selection and integration. The background to the method and its evolution with experience can be found in several publications (Sutcliffe & de Angeli, 2005; Faraday & Sutcliffe, 1996, 1997b, 1998b; Sutcliffe & Faraday, 1994). A more detailed description is given in Sutcliffe (2003). The time-to-market pressure gives little incentive for design, so at first reading, a systematic approach may seem to be counter to the commercial drivers of development; however, I would argue that if multimedia design does not adopt a usability engineering approach, it will fail to deliver effective and usable products.

Multimedia applications have significant markets in education and training, although dialogue in many systems is restricted to drill and quiz interaction and simple navigation. This approach, however, is oversimplified: for training and education, interactive simulations and microworlds are more effective (Rogers & Scaife, 1998). Multimedia has been used extensively in task-based applications in process control and safety critical systems (Alty, 1991; Hollan, Hutchins, & Weitzman, 1984); however, most transaction processing applications are currently treated as standard interfaces rather than multimedia-based designs. With the advent of the web and e-commerce, this view may change.

Design issues for multimedia user interfaces (UIs) expand conventional definitions of usability (e.g., International Standards Organization, 1997) into five components:

1. *Operational usability* is the conventional sense of usability that concerns design of graphical user interface features such as menus, icons, metaphors, and navigation in hypermedia.
2. *Information delivery* is a prime concern for multimedia or any information-intensive application, and raises issues of media selection, integration, and design for attention.
3. *Learning:* Training and education are both important markets for multimedia; hence, learnability of the product and its content are key quality attributes. Design of educational technology, however, is a complex subject in its own right, and multimedia is only one part of the design problem.
4. *Utility:* In some applications, utility will be the functionality that supports the user's task; in others, information delivery and learning will represent the value perceived by the user.
5. *Engagement and attractiveness:* The attractiveness of multimedia is now a key factor, especially for websites. Multimedia interfaces have to attract users and deliver a stimulating user experience, as well as being easy to use and learn.

Multimedia design involves several specialisms that are technical subjects in their own right. For instance, design of text is the science (or art) of calligraphy that has developed new fonts over many years; visualization design encompasses the creation of images, either drawn or captured as photographs. Design of moving images, cartoons, video, and film are further specializations, as are musical composition and design of sound effects. Multimedia design lies on an interesting cultural boundary between the creative artistic community and science-based engineering. One implication of this cultural collision is that space precludes "within media" design, or guidelines for design of one particular medium, which is dealt with in depth in this chapter. Successful multimedia design often requires teams of specialists who contribute from their own skill sets (Kristof & Satran, 1995; Mullet & Sano, 1995).

DEFINITIONS AND TERMINOLOGY

Multimedia essentially extends the graphical user interface (GUI) paradigm by providing a richer means of representing information for the user by use of image, video, sound, and speech. Some views of what constitutes multimedia can be found in Bernsen (1994), who proposed a taxonomy of analog versus discrete media, which he called modalities, as well as visual, audio, and tactile dimensions. Heller and Martin (1995) took a more conventional view of classifying image, text, video, and graphics for educational purposes. The following definitions broadly follow those in the ISO standard 14915 on Multimedia User Interface Design (International Standards Organization, 1998). The starting point is to ask about the difference between what is perceived by someone and what is stored on a machine.

Communication concepts in multimedia can be separated into

- *Message:* The content of communication between a sender and receiver.
- *Medium* (plural, *media*)*:* The means by which that content is delivered. Note that this is how the message is represented rather than the technology for storing or delivering a message. There is a distinction between perceived media and physical media such as CD-ROM, hard disk, and so forth.
- *Modality:* The sense by which a message is sent or received by people or machines. This refers to the senses of vision, hearing, touch, smell, and taste.

A message is conveyed by a medium and received through a modality. A modality is the sensory channel that we use to send and receive messages to and from the world—essentially, our senses. Two principal modalities are used in human-computer communication:

1. *Vision:* All information received through our eyes, including text and image-based media.
2. *Hearing:* All information received through our ears, as sound, music, and speech.

In the future, as multimedia converges with Virtual Reality (VR), we will use other modalities more frequently: *haptic* (sense of touch), *kinaesthetic* (sense of body posture and balance, *gustation* (taste) and *olfaction* (smell). These issues are dealt with in chapters 21, Multimodal interfaces (Oviatt, Darves, & Coulston, 2004), and 31, Virtual environments (Stanney).

Defining a medium is not simple because it depends on how it was captured in the first place, how it was designed, and how it has been stored. For example, a photograph can be taken on film, developed, and then scanned into a computer as a digitized image. The same image may have been captured directly by a digital camera and sent to a computer as an e-mail file. At the physical level, media may be stored by different techniques.

Physical media storage has usability implications for the quality of image and response time in networked multimedia. A screen image with 640×480 VGA resolution using 24 bits per pixel for good color coding gives 921,600 bytes, so at 30 frames per second, one second needs around 25 megabytes of memory or disk space. Compression algorithms, for instance, MPEG (Moving Pictures Expert Group), reduce this by a factor of 10. Even so, storing more than a few minutes of moving image consumes megabytes. The usability trade-off is between the size of the display footprint (e.g., window size), the resolution measured in dots per inch, and the frame rate. The ideal might be full-screen high resolution (600 dpi) at 30 frames per second; with current technology, a 10-cm window at 300 dpi and 15 frames per second is more realistic. Physical image media constraints become more important on networks, when bandwidth will limit the desired display quality. Sound, in comparison, is less of a problem. Storage demands depend on the fidelity required for replay. Full stereo with a complete range of harmonic frequencies consumes only 100 kilobytes for five minutes, so there are few technology constraints on delivery of high-quality audio.

COGNITIVE BACKGROUND

The purpose of this section is to give a brief overview of cognitive psychology as it affects multimedia design. More details can be found in section II, Humans in HCI.

Perception and Comprehension

Generally, our eyes are drawn to moving shapes, then complex, different, and colorful objects. Visual comprehension can be summarized as "what you see depends on what you look at and what you know." Multimedia designers can influence what users look at by controlling attention with display techniques such as use of movement, highlighting and salient icons; however, designers should be aware that the information people assimilate from an image also depends on their internal motivations, what they want to find and how well they know the domain (Treisman, 1988). A novice will not see interesting plant species in a tropical jungle, whereas a trained botanist will. Selection of visual content therefore has to take the user's knowledge and task into account.

Because the visual sense receives information continuously, it is overwritten in working memory (Baddeley, 1986). This means that memorization of visually transmitted information is not always effective unless users are given time to view and comprehend images. Furthermore, users only extract very high-level or "gist" (general sense) information from moving images. Interpreting visual information relies on memory. In realistic images, this process is automatic; however, with nonrealistic images, we have to think carefully about the meaning, for example, to interpret a diagram. While extraction of information from images is rapid, it does vary according to the complexity of the image and how much we know about the domain. Sound is a transient medium, so unless it is processed quickly, the message can be lost. Even though people are remarkably effective at comprehending spoken language and can interpret other sounds quickly, the audio medium is prone to interference because other sounds can compete with the principal message. Because sound is transient, information in speech will not be assimilated in detail, and so only the gist will be memorized (Gardiner & Christie, 1987).

Selective Attention

We can only attend to a limited number of inputs at once. While people are remarkably good at integrating information received by different senses (e.g., watching a film and listening to the sound track), limits are determined by the psychology of human information processing (Wickens, Sandry, & Vidulich, 1983). Our attention is selective and closely related to perception; for instance, we can overhear a conversation in a room with many people speaking (the cocktail party effect). Furthermore, selective attention differs between individuals and can be improved by learning. For example, a conductor can distinguish the different instruments in an orchestra, whereas a typical listener cannot. However, all users have cognitive resource limitations, which means that information delivered on different modalities (e.g., by vision and sound) have to compete for the same resource. For instance, speech and printed text both require a language-understanding resource, while video and a still image use image-interpretation resources. Cognitive models of information processing architectures (e.g., Interacting Cognitive Subsystems; Barnard, 1985) can show that certain media combinations will not result in effective comprehension because they compete for the same cognitive resources, thus creating a processing bottleneck. We have two main perceptual channels for receiving information: (a) vision and (b) hearing. Information going into these channels has to be comprehended before it can be used. Figure 20.1, based on Card, Moran, and Newell's (1983) Model Human Processor, shows the cognitive architecture of human information processing and resource limitations that lead to multimedia usability problems.

Capacity overflow may happen when too much information is presented in a short period, swamping the user's limited working memory and cognitive processor's capability to comprehend, chunk, and then memorize or use the information. The connotation is to give users control over the pace of information delivery. Integration problems arise when the message on two media is different, making integration into working

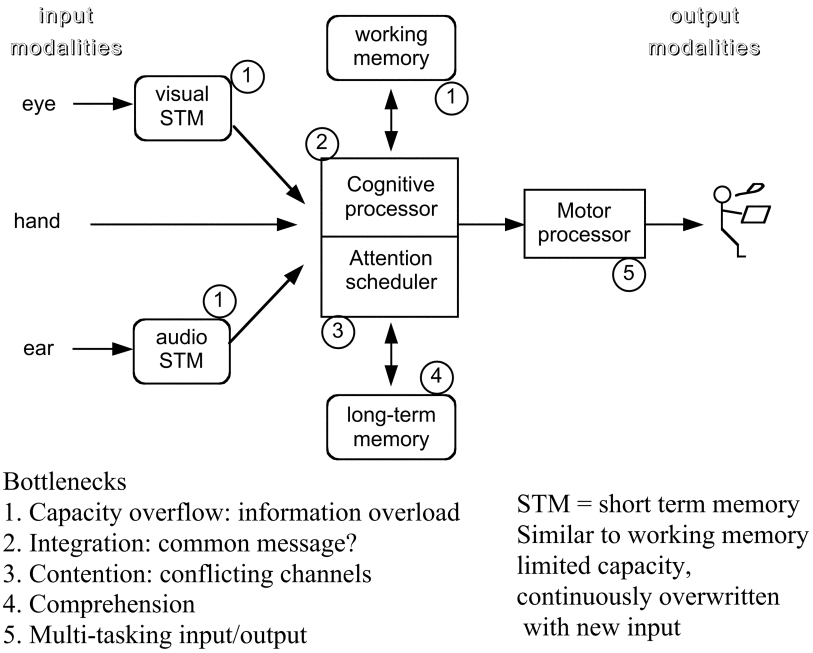


FIGURE 20.1. Approximate model of human information processing using a "human as computer system" analogy, based on the Model Human Processor (Card, Moran, & Newell, 1983; for more on cognitive models, see Proctor & Vu, chap. 2, this Handbook; Byrne, chap. 5, this Handbook)

memory difficult; this leads to the thematic congruence principle. Contention problems are caused by conflicting attention between dynamic media, and when two inputs compete for the same cognitive resources, e.g., speech and text both requiring language understanding at once. Comprehension is related to congruence; we understand the world by making sense of it with our existing long-term memory. Consequently, if multimedia content is unfamiliar, we cannot make sense of it. Finally, multitasking makes further demands on our cognitive processing, so we will experience difficulty in attending to multimedia input while performing output tasks.

Making clear a theme in a multimedia presentation involves directing the user's reading and viewing sequence across different media segments. Video and speech are processed in sequence, while text enforces a serial reading order by the syntactic convention of language. In contrast, viewing image media is less predictable since it depends on the size and complexity of the image, the user's knowledge of the contents, task, and motivation (Norman & Shallice, 1986), and the designed effects for salience. Attention-directing effects can increase the probability that the user will attend to an image component, although no guarantee can be given that a component will be perceived or understood.

Emotion and Arousal

The content of image media in particular can evoke an emotional response, which can be used to promote a more exciting and engaging user experience (e.g., the use of human image and speech to persuade users; see chap. 7, this Handbook). These issues are dealt with more extensively in other chapters. People treat human photographs, video and even animated characters with similar social responses as they give to real people, so human image content can be used to increase interest and draw attention. Emotional responses (see chap. 4) can be invoked not only by content, but also by surprise interactive effects (e.g., when a character suddenly appears to challenge the users). Surprise effects, moving image, and unexpected stimulating images all affect the arousal system, which broadly equates with our feeling of excitement. Designs that stimulate our arousal are more likely to be memorable and engaging.

Learning and Memorization

Learning is the prime objective in tutorial multimedia. In these applications, the objective is to create a rich memory schema, which can be accessed easily in the future. We learn more effectively by active problem solving, or learning by doing. This approach is at the heart of constructivist learning theory (Papert, 1980), which has connotations for tutorial multimedia. Interactive microworlds where users learn by interacting with simulations, or constructing and testing the simulation, give a more vivid experience that forms better memories (Rogers & Scaife, 1998). Multiple viewpoints help to develop rich schemata by presenting different aspects of the same problem, so the whole concept can be integrated from its parts. An example might be to explain the structure of an engine, then how it operates, and finally, display a causal model of why it works. Schema integration during memorization fits the separate viewpoints together.

The implications from psychology are summarized in the form of multimedia design principles that amplify and extend

those proposed for general user interface design (e.g., ISO 9241 part 10; International Standards Organization, 1997). The principles are high-level concepts that are useful for general guidance, but they have to be interpreted in a context to give more specific advice.

- *Thematic congruence:* Messages presented in different media should be linked together to form a coherent whole. This helps comprehension as the different parts of the message make sense when fitted together. Congruence is partly a matter of designing the content so it follows a logical theme—for example, the script or story line makes sense and does not assume too much about the user's domain knowledge—and partly a matter of attentional design to help the user follow the message thread across different media (Ayers & Sweller, 2005).
- *Manageable information loading:* Messages presented in multimedia should be delivered either at a pace that is under the user's control, or at a rate that allows for effective assimilation of information without causing fatigue. The rate of information delivery depends on the quantity and complexity of information in the message, the effectiveness of the design in helping the user extract the message from the media, as well as the user's domain knowledge and motivation. Two ways to reduce information overload are to (a) avoid excessive use of concurrent dynamic media and (b) give the user time to assimilate complex messages.
- *Ensure compatibility with the user's understanding:* Media should be selected that convey the content in a manner compatible with the user's existing knowledge (e.g., the radiation symbol and road sign icons are used to convey hazards and dangers to users who have the appropriate knowledge and cultural background). The user's ability to understand the message is important for designed image media (e.g., diagrams, graphs) when interpretation is dependent on the user's knowledge and background.
- *Complementary viewpoints:* Similar aspects of the same subject matter should be presented on different media to create an integrated whole. Showing different aspects of the same object—for example, picture and design diagram of a ship—can help memorization by developing richer schema and better memory cues.
- *Consistency:* Consistency helps users learn an interface by making the controls, command names, and layout follow a familiar pattern. People recognize patterns automatically, so operating the interface becomes an automatic skill. Consistent use of media to deliver messages of a specific type can help by cueing users with what to expect.
- *Interaction and engagement:* Interaction and engagement help understanding and learning by encouraging the user to problem solve. Memory is an active process. Interaction increases arousal and this make the user's experience more vivid, exciting, and memorable.
- *Reinforce messages:* Redundant communication of the same message on different media can help learning. Presentation of the same or similar aspects of a message helps memorization by the frequency effect. Exposing users to the same thing in a different modality also promotes rich memory cues.

DESIGN PROCESS

Multimedia design has to address the problems inherent in the design of any user interface, viz. defining user requirements, tasks, dialogue design; however, three issues specifically concern multimedia:

1. *Matching the media to the message* by selecting and integrating media so the user comprehends the information content effectively.
2. *Managing users' attention* so key items in the content are noticed and understood, and the user follows the message thread across several media.
3. *Interaction and navigation* so the user can access, play, and interact with media in an engaging and predictable manner.

Figure 20.2 gives an overview of the design process that addresses these issues.

The method shown in the figure starts by requirements and information analysis to establish the necessary content and communication goals of the application. It then progresses to domain and user characteristic analysis to establish a profile of the user and the system environment. The output from these stages feeds into media selection and integration, which match the logical specification of the content to available media resources. This is interleaved with interaction design, unless the application is restricted to information presentation. Design then progresses to thematic integration of the user's reading/viewing sequence and design to direct the users' attention. The method can be tailored to fit within different development approaches. For

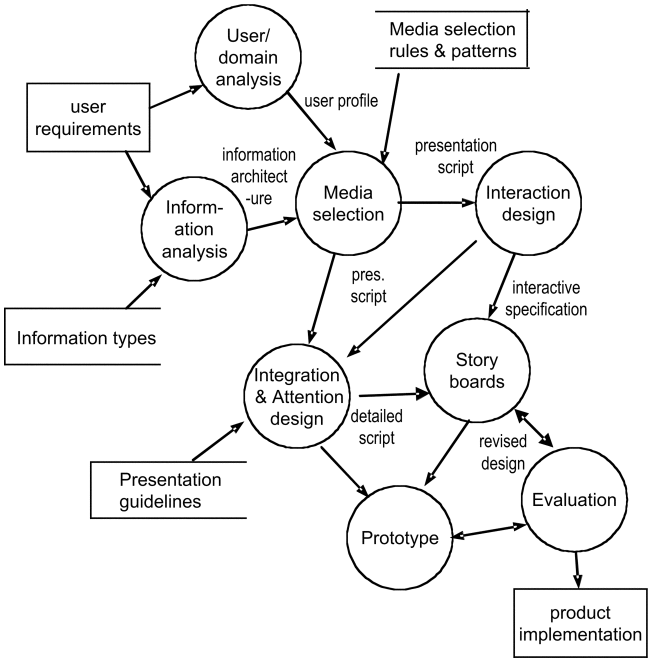


FIGURE 20.2. Overview of the multimedia design process expressed as a data flow diagram.

instance, in rapid application development, storyboards, prototypes, and iterative build-and-evaluate cycles would be used. On the other hand, in a more systematic, software engineering approach, more detailed specifications and scripts will be produced before design commences. Even though the process is described as a sequence, in practice, the stages are interleaved and iterated; however, requirements, information modeling, and media selection should be carried out, even if they are not complete, before subsequent design stages commence.

Design approaches in multimedia tend to be iterative and user-centered. *Storyboards* are a well-known means of informal modeling in multimedia design (Nielsen, 1995; Sutcliffe, 1999). Originating from animation and cartoon design, storyboards are a set of images which represent key steps in a design. Translated to software, storyboards depict key stages in interaction and are used for conducting walkthroughs to explain what happens at each stage. Allowing the users to edit storyboards and giving them a construction kit to build their own encourages active participation. Storyboards are followed by building concept demonstrators using multimedia authoring tools (e.g., Macromedia Director, Toolbook) to rapidly develop early prototypes. *Concept demonstrators* are active simulations that follow a scenario script of interaction; departure from the preset sequence is not allowed. Several variations can be run to support comparison; however, the user experience is passive. In contrast, users can test *interactive prototypes* by running different commands or functions. The degree of interactivity depends on the implementation cost which increases as prototypes converge with a fully functional product.

Users, Requirements, and Domains

The starting point for multimedia, as in all applications, is requirements analysis. The difference in multimedia lies in the greater emphasis on information requirements. A variety of analytic approaches can be adopted, such as task-analysis (see Redish, chap. 48, this Handbook), contextual inquiry (Holtzblatt, chap. 49, this Handbook) or scenario analysis (Rosson & Carroll, chap. 53, this Handbook). Requirements are listed and categorized into information, task-related, and nonfunctional classes. These will be expanded in subsequent analyses.

It is important to get a profile of the target user population to guide media selection. There are three motivations for user analysis:

1. *Choice of modalities*, which is important for people with disabilities, but also for user preferences. Some people prefer verbal-linguistic material over image.
2. *Tuning the content* presented to the level of users' existing knowledge, which is particularly important for training and educational applications.
3. *Capturing users' expectations* so the experience can be geared toward their backgrounds (e.g., different styles for younger, older people, cultures, and socio-economic audiences).

Acquiring information about the level of experience possessed by the potential user population is important for customization. User profiles are used to design training applications to ensure that the right level of tutorial support is provided, and to assess the users' domain knowledge so that appropriate media can be selected. This is particularly important when symbols, designed images, and diagrammatic notations may be involved. The roles and backgrounds of users will have an important bearing on design. For example, marketing applications will need simple, focused content and more aesthetic design, whereas tutorial systems need to deliver detailed content. Information kiosk applications need to provide information, as do task-based applications, but decision support and persuasive systems (Fogg, 1998; see also chap. 7, this Handbook) also need to ensure users comprehend and are convinced by messages. Domain knowledge, including use of conventions, symbols, and terminology in the domain, is important because less experienced users will require that more complete information is presented.

The context and environment of a system will also have an important bearing on design. For example, tourist information systems in outdoor public areas will experience a wide range of lighting conditions, which can make image and text hard to read. High levels of ambient noise in public places or factory floors can make audio and speech useless. Hence, it is important to gather information on (a) the location of use (e.g., office, factory floor, public/private space, and hazardous locations), (b) pertinent environmental variables (e.g., ambient light, noise levels, and temperature), (c) usage conditions (e.g., single user, shared use, and broadcast) and (d) expected range of locations (e.g., countries, languages, and cultures). Choice of language, icon conventions, interpretation of diagrams, and choice of content all have a bearing on design of international user interfaces.

As well as gathering general information about the system's context of use, domain modeling can prove useful for creating the system metaphor. Domain models are recorded as sketches of the work environment, showing the layout and location of significant objects and artifacts, accompanied by lists of environmental factors. Structural metaphors for organizing information and operational metaphors for controls and devices have their origins in domain analysis.

Information Architecture

This activity consists of several activities that will differ according to the type of application. Some applications might have a strong task model (e.g., a multimedia process control application where the tasks are monitoring a chemical plant, diagnosing problems, and supporting the operator in controlling plant operation). In task-driven applications, information requirements are derived from the task model. In information provision applications, such as websites with informative roles, information analysis involves categorization, and the architecture generally follows a hierarchical model. In the third class of explanatory or thematic applications, analysis is concerned with the story or argument (e.g., how the information should be explained or delivered). Educational multimedia and websites with persuasive missions fall into the last category.

In task-driven applications, information needs are annotated on the task model following a walkthrough that asks users what information they need to (a) complete the task subgoal, (b) take a decision at this step, or (c) provide as input (see Sutcliffe,

1997). In information provision applications, classification of the content according to one of more user views defines the information architecture; for example, most university departments have an information structure with upper-level categories for research, undergraduate courses, postgraduate courses, staff interests, departmental organization, mission and objectives, and so forth. A theme or story line should be developed for explanatory applications. This will depend on the application's objectives and the message the owner wishes to deliver. An example thematic map from a hypothetical health awareness application is illustrated in Fig. 20.3.

The requirement is to convince people of the dangers of heart disease. The theme is a persuasive argument that first tries to convince people of the dangers (e.g, smoking, poor diet, stressed lifestyles, etc.), and then convince them of ways to improve their lifestyles to prevent heart disease, followed by reinforcing the message with the benefits of a healthy lifestyle (e.g., lower health insurance, saving money, longer life, etc.). Subthemes are embedded at different points so users can explore the facts behind heart disease, the statistics and their exposure, how to get help, and so forth. Information is then gathered for each node in the thematic map. How this architecture will be delivered depends on interaction design decisions: it could become an interactive story to explore different lifestyle choices, combined with a quiz. The outcome of information architecture analysis will be an information-enhanced task model, a thematic map, or a hierarchy/network to show the structure and relationships of information categories. The next step is to analyze the information content by classifying it by types.

Information types are amodal, conceptual descriptions of information components that elaborate the content definition. Information types specify the message to be delivered in a multimedia application and are operated on by mapping rules to select appropriate media resources. The following definitions are based on the Task-Based Information Analysis Method (Sutcliffe, 1997) and ISO 14915, part 3 (International Standards Organization, 2000).

The information types are used in "walkthroughs", in which the analyst progresses through the task/scenario/use case asking questions about information needs. This can be integrated with data modeling (or object/class modeling), so that the information in objects and their attributes can be categorized by the following types, using the decision tree shown in Fig. 20.4. The first question is whether information represents concrete facts about the real world or more abstract, conceptual information; this is followed by questions about the information that relates to change in the world or describes permanent states. Finally, the decision tree gives a set of ontological categories to classify information that expand on type definitions commonly found in software engineering specifications. More complex ontologies are available (Arens, Hovy, & Van Mulken, 1993; Mann & Thompson, 1988), so the classification presented in Fig. 20.4 is a compromise between complexity and ease of use. A finer-grained classification enables more finely tuned media selection decisions, but at the cost of more analysis effort.

Components are classified by "walking through" the decision tree using the definitions and the following questions:

- Is the information contained in the component physical or conceptual?
- Is the information static or dynamic—for example, does it relate to change or not?
- Which type in the terminal branch of the tree does the information component belong to?

It is important to note that one component may be classified with more than one type; for instance, instructions on how to get to the railway station may contain procedural information (the instructions "turn left," "straight ahead," etc.), and spatial or descriptive information ("the station is in the corner of the square, painted blue"). The information types are "tools for thought" and can be used either to classify specifications of content or to consider what content may be necessary. To illustrate, for the task "navigate to the railway station," the content may be minimally specified as "instructions how to get there," in which case the information types prompt questions in the form "what sort of information does the user need to fulfil the task/user goal?" Alternatively, the content may be specified as a scenario narrative of directions, waymarks to recognize and description of the target. In this case, the types classify components in the

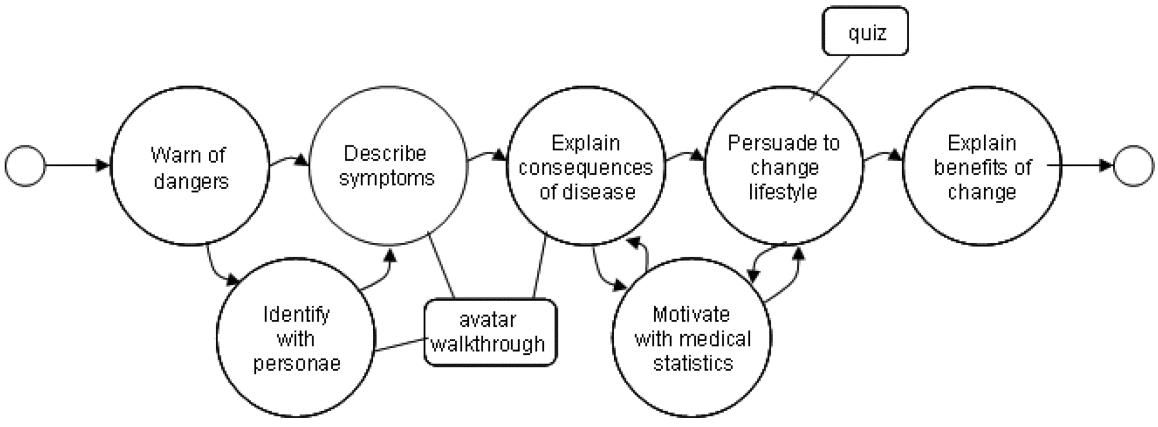


FIGURE 20.3. Thematic map for a healthcare promotion application.

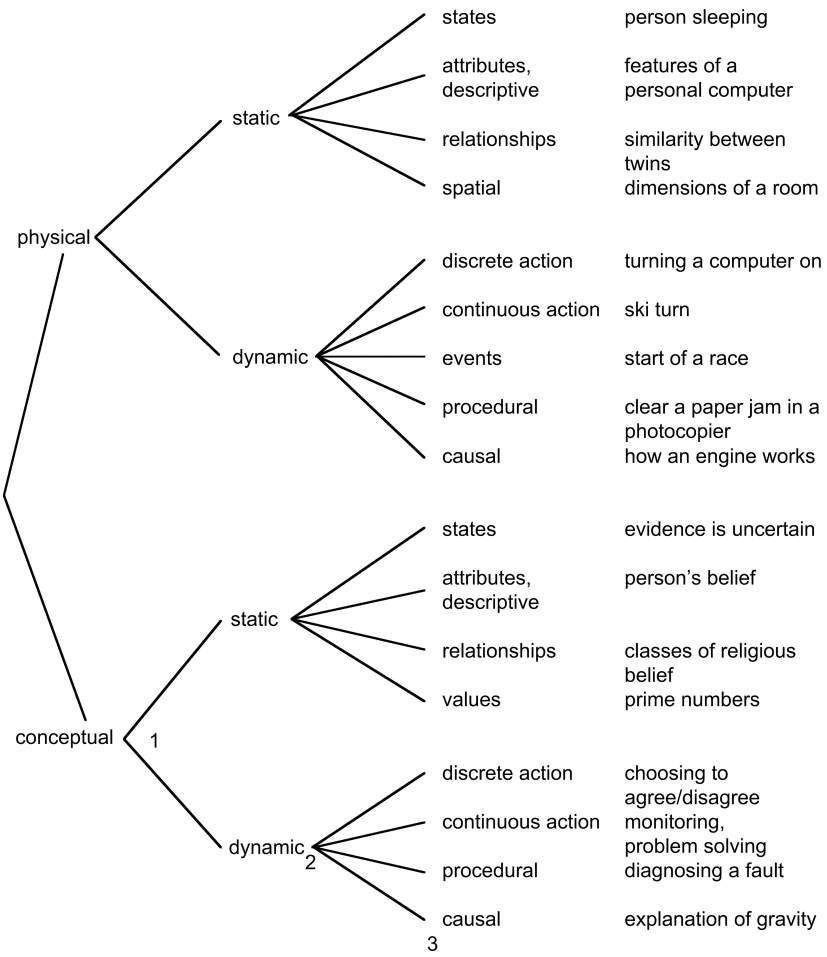


FIGURE 20.4. Decision tree for classifying information types. The first decision point reflects abstraction from the real world, the second points to change in time and the third categorizes content.

narrative to elucidate the deeper structure of the content. The granularity of components is a matter of the designer's choice and will depend on the level of detail demanded by the application. To illustrate the analysis,

Communication goal: Explain how to assemble a bookshelf from ready-made parts.

Information component 1:
- Parts of the bookshelf: sides, back, shelves, connecting screws
- Mapping to information types:
  - Physical-Static-Descriptive; parts of the bookshelf are tangible, don't change, and need to be described
  - Physical-Static-Spatial; dimensions of the parts, how they are organized
  - Physical-Static-Relationship type could also be added to describe which parts fit together

Information component 2:
- How to assemble parts instructions
- Mapping to information types:
  - Physical-Dynamic-Discrete action
  - Physical-Dynamic-Procedure
  - Physical-Static-State; to show final assembled bookshelf.

Media Selection and Combination

The information types are used to select the appropriate media resource(s). Media classifications have had many interpretations (Alty, 1997; Bernsen, 1994; Heller & Martin, 1995). The following classification focuses on the psychological properties of the representations, rather than the physical nature of the medium (e.g., digital or analog encoding in video). Note that these definitions are combined to describe any specific medium, so speech is classified as an audio, linguistic medium, while a cartoon is classified as a nonrealistic (designed) moving image.

The definitions may be usefully considered in two dimensions of abstraction: (a) the designer's involvement in creating the medium, and (b) the rate of change. Media resources are classified using the decision tree illustrated in Fig. 20.5. More

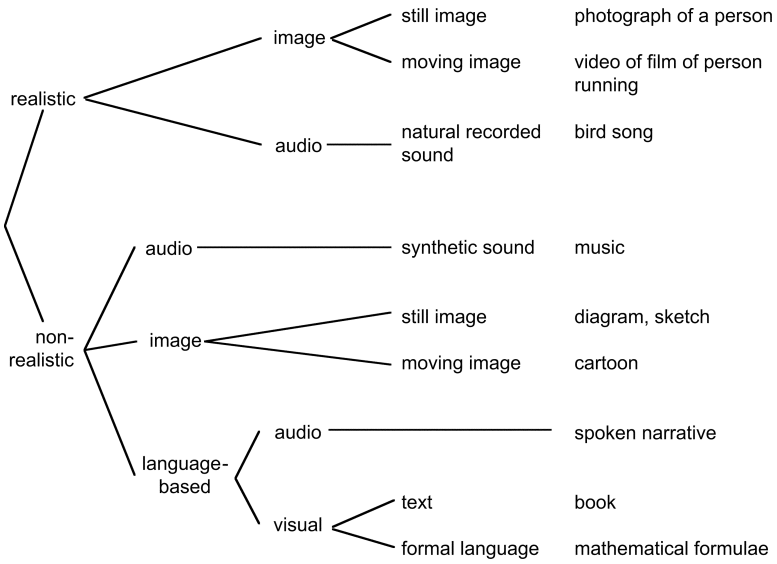


FIGURE 20.5. Decision tree for classifying media resources.

fine-grained, taxonomic distinctions can be made, for instance, between different signs and symbolic languages (see Bernsen, 1994), but as with information types, richer taxonomies increase specification effort.

The approach to classifying media uses a walkthrough of the decision tree with the following questions that reflect the facets of the classification:

- Is the medium perceived to be realistic or not? Media resources captured directly from the real world will usually be realistic (e.g., photographs of landscapes, bird song sound recordings). In contrast, nonrealistic media are created by human action; however, the boundary case category that illustrates the dimension is a realistic painting of a landscape.
- Does the medium change over time or not? The boundary case here is the rate of change, particularly in animations where some people might judge ten frames/sec to be a video, but a PowerPoint presentation of five slides in one minute to be a sequence of static images.
- Which modality does the resource belong to? In this case, the categories are orthogonal, although one resource may exhibit two modalities (e.g., a film with a sound track communicates in both visual and audio modalities).

Classification of media resources facilitates mapping of information types to media resources; however, the process of selection may also guide the acquisition or creation of appropriate media resources. So if the selection process indicates the need for a resource that is not in the designer's media resource library, the classification guides the necessary acquisition/creation. Cost trade-offs will naturally be considered in this process. Finally, the classification provides a mechanism for indexing media resource libraries.

Media selection. Recommendations for selecting and influencing the user's attention have to be interpreted according to the user's task and design goal. If information provision is the main design goal (e.g., a tourist kiosk information system), then persistence of information and drawing attention to specific items is not necessarily as critical as in tutorial applications. Task and user characteristics influence media choice; for instance, verbal media are more appropriate to language-based and logical reasoning tasks, while visual media are suitable for spatial tasks involving moving, positioning, and orienting objects. Some users may prefer visual media, while image is of little use for blind users. Media resources may be available for selection, or have to be purchased from elsewhere. If existing media can be edited and reused, this is usually preferable to creating new media from scratch. Graphical images can be particularly expensive to draw, whereas capture of images by scanning is usually quick and cheap.

The suggested mappings from information types to media resources are given in Table 20.1. Information types are mapped to a simple categorization of media as (a) static, nonstreaming media; (b) still image, text, diagrams, graphs and dynamic media; (c) audio: natural sounds, speech, music, and designed sounds; and (d) video and animation. Several media map to each type, although the preferred choice is shown in italics. Since most components in the information architecture will have multiple information types and each information type maps to several media, the selection process encourages multimedia integration. For example, when a procedure for explaining a physical task is required, first a series of realistic images will be selected, followed by video and speech to integrate the steps, then text to summarize the key points. The mappings are supplemented by the following heuristics:

- To convey detail, use static media (e.g., text for language-based content, diagrams for models, or still image for physical detail of objects; Booher, 1975; Faraday & Sutcliffe, 1998b).
- To engage the user and draw attention, use dynamic media (e.g., video for physical information, animation, or speech).

TABLE 20.1. Media Selection Example

Information Type Media Type	Causation	Conceptual	Continuous Action	Descriptive	Discrete Action	Event	Physical	Procedure	Relationship	Spatial Information	State	Value
Realistic audio	Sound of rain and storms		Sound of skiing		Click of ON switch	Sound of the starting gun	*Noise of a tornado*			Echoes in a cave	Sound of snoring	Musical note encodes a value
Nonrealistic audio		Rising tone illustrates increasing magnetic force	Continuous tone signals progress of action	Morse code describes a ship	Tones signal open/close door	*Alarm siren*			Tones associate two objects	Sonar and Doppler effect	Continuous sound in a heart beat monitor	
Speech	*Tell someone why El Nino happens*	Tell someone about your religious beliefs	Tell someone what a ski turn looks like	Verbal description of a person	Tell someone how to turn computer on	*Tell someone race has started*	Tell someone how it feels to be in a storm	Speak instructions on engine assembly	Tell someone Jack and Jill are related	Tell someone pathway to and location of railway station	Tell someone "Jane's asleep"	Verbal report of numbers, figures
Realistic still image	*Photograph of El Nino storms and ocean currents*	Statue of Liberty photograph represents "freedom"	Set of photographs showing snap shots of action	*Overview and detail photographs of a car*	*Photograph of computer* ON *switch*	Photograph of the start of a race	*Photograph of a person's face*	*Photographs showing engine assembly*		*Photograph of a landscape*	Photograph of a person sleeping	
Nonrealistic still image	*Diagrams of ocean currents and sea temp. to explain El Nino*	*Hierarchy diagram of plant taxonomy*	Diagram with arrow depicting ski turn motion	Histogram of ageing population	Diagram showing where and how to press ON switch	Event symbol in a race sequence diagram		Explode parts diagram of engine with assembly numbers	*Graphs, histograms,* ER *diagrams*	*Map of the landscape*	Waiting state symbol in race sequence diagram	*Charts, graphs, scatter plots*
Text	*Describe reasons for El Nino storms*	*Explain taxonomy of animals*	Describe ski turn action	*Describe a person's appearance*	Describe how to turn computer on	Report that the race has started	Report of the storm's properties	*Bullet point steps in assembling engine*	*Describe brother and sister relationship*	Describe dimensions of a room	*Report that the person is asleep*	*Written number one, two*
Realistic moving image	*Video of El Nino storms and ocean currents*		*Movie of person turning while skiing*	Aircraft flying		*Movie of the start of a race*	*Movie of a storm*	*Video of engine assembly sequence*		Fly through landscape	Video of a person sleeping	
Nonrealistic moving image	*Animation of ocean temperature change and current reversal*	Animated diagram of force of gravity	Animated mannequin doing ski turn		Animation showing operation of ON switch	Animation of start event symbol in diagram		*Animation of parts diagram in assembly sequence*	Animation of links on ER diagram			
Language-based: formal, numeric	Equations, functions formalizing cause and effect	*Symbols denoting concepts, e.g., pi*			Finite state automata	Event based notations		Procedural logics, process algebras	Functions, equations, grammars		State based languages, e.g., Z	Numeric symbols

Note: The italics denote the preferred mappings for media and information types, while ordinary text shows other potential media uses for the information type.

- For spatial information, use diagrams, maps with photographic images to illustrate detail, animations to indicate pathways (Bieger & Glock, 1984; May & Barnard, 1995).
- For values and quantitative information, use charts and graphs for overviews and trends, supplemented by tables for detail (Bertin, 1983; Tufte, 1997).
- Abstract concepts, relationships, and models should be illustrated with diagrams explained by text captions and speech to give supplementary information.
- Complex actions and procedures should be illustrated as a slideshow of images for each step, followed by a video of the whole sequence to integrate the steps. Text captions on the still images and speech commentary provide supplementary information (Hegarty & Just, 1993). Text and bullet points summarize steps at the end, so choice trade-offs may be constrained by cost and quality considerations.
- To explain causality, still and moving image media need to be combined with text (e.g., the cause of a flood is explained by text describing excessive rainfall with an animation of the river level rising and overflowing its banks; Narayanan & Hegarty, 1998, 2002). Causal explanations of physical phenomena may be given by introducing the topic using linguistic media, showing the cause and effect by a combination of still image and text with speech captions for commentary; integrating the message by moving image with voice commentary and providing a bullet point text summary.

The end point of media selection is media integration: one or more media will be selected for each information group to present complementary aspects of the topic. Some examples of media combination that amplify the basic selection guidelines are given in Table 20.1. The table summarizes the media selection and combinations for each information type; the first preference media choice for each is shown in italics.

Engagement and attractiveness. The previous guidelines were oriented to a task-driven view of media. Media selection, however, can also be motivated by aesthetic choice. These considerations may contradict some of the earlier guidelines because the design objective is to please the user and capture the user's attention, rather than deliver information effectively. First, a health warning should be noted: the old saying "beauty is in the eye of the beholder" has good foundation. Judgments of aesthetic quality suffer from considerable individual differences. A person's reaction to a design is a function of their motivation (see Brave & Nass, chap. 4, this Handbook), individual preferences, knowledge of the domain, and exposure to similar examples, to say nothing of peer opinion and "fashion." Furthermore, attractiveness is often influenced more by content than the choice of media or presentation format. The following heuristics should therefore be interpreted with care and their design manifestations tested with users:

- *Dynamic media*, especially video, have an arousing effect and attract attention; hence, video and animation are useful in improving the attractiveness of presentations. Animation must be used with care, however, as gratuitous video that cannot be turned off quickly offends (Spool, Scanlon, Snyder, Schroeder, & de Angelo, 1999).
- *Speech* engages attention because we naturally listen to conversation. Choice of voice depends on the application: female voices for more restful and information effects, male voices to suggest authority and respect (Reeves & Nass, 1996).
- *Image* must be selected with careful consideration of content. Images may be selected for aesthetic motivations, to provide a restful setting for more important foreground information (Mullet & Sano, 1995). Backgrounds in half shades and low saturation color provide more depth and interest in an image.
- *Music* has an important emotive appeal, but it must be used with care. Classical music may be counterproductive for a younger audience, while older listeners will not find heavy metal pop attractive.
- *Natural sounds* such as running water, wind in trees, bird songs, and waves on a seashore have restful properties and hence decrease arousal.

Media integration rules may also be broken for aesthetic reasons. If information transfer is not at a premium, use of two concurrent video streams might be arousing for a younger audience, as MTV (music television) and pop videos indicate. Multiple audio and speech tracks can give the impression of complex, busy, and interesting environments.

Aesthetic design. If the user profile and domain analysis indicate that having a pleasing and attractive design is important for the user's perception, then aesthetics need to be considered in depth; however, aesthetics should be considered as a design criterion for all applications, since poor appearance and interaction design may provoke adverse reaction (Norman, 2004). Some studies suggest that aesthetic design is an important component of usability and overall preference (Tractinsky, 1997; Tractinsky, Shoval-Katz, & Ikar, 2000); however, others have shown that aesthetic preferences are open to contextual effects on users' judgment (Sutcliffe & de Angeli, 2005). Judging when aesthetics may be important is not easy. For example, in e-commerce applications with high-value, designer-label products, aesthetic presentation is advisable, as is similar when selling to a design-oriented audience; however, the decision is not clear-cut in many applications.

Aesthetic design primarily concerns graphics and visual media. Evaluation questionnaires assess design on classic aesthetics, which broadly equate with conventional usability guidelines on structured and consistent layout, and expressive aesthetics that capture the more creative aspects of visual design (Lavie & Tractinsky, 2004); however, these measure user reaction to general design aspects such as "original, fascinating, clear, and pleasant." The following heuristics provide more design-directed guidance, but they may also be employed for evaluation (Sutcliffe, 2002; Sutcliffe & de Angeli, 2005):

- *Judicious use of color:* Color use should be balanced, and low saturation pastel colors should be used for backgrounds. Designs should not use more than two to three fully saturated intense colors.
- *Symmetry:* Symmetrical layouts (e.g., bilateral, radial organization) that can be folded over to show the symmetrical match.

- *Simplicity and space:* Uncluttered, simple layouts that use space to separate and emphasize key components.
- *Shape:* Curved shapes convey an attractive visual style when contrasted with rectangles.
- *Structured and consistent layout:* Use of grids to structure image components and promote consistency between pages.
- *Depth of field:* Use of layers in an image and washed out background images stimulates curiosity and can be attractive by promoting a peaceful effect.
- *Design of unusual or challenging images* that stimulate the users' imagination and increase attraction: Unusual images often disobey normal laws of form and perspective.

Several sources provide more detailed advice on aesthetics and visual design (Kristof & Satran, 1995; Mullet & Sano, 1995); however, advice is usually given as examples of good design rather than specific guidelines.

Image and identity. Design of media for motivation is a complex area in its own right, and this topic is dealt with in more depth in Fogg (chap. 7, this Handbook), so the treatment here will focus on media selection issues.

Simple photographs or more complex interactive animations (talking heads or full body mannequins) have an attractive effect. We appear to ascribe human properties to computers when interfaces give humanlike visual cues (Reeves & Nass, 1996); however, the effectiveness of media representing people depends on the characters' appearance and voice (see Fig. 20.6, courtesy of Sapient). In human-human conversation, we modify our reactions according to our knowledge, or assumptions about, the other person's role, group identification, culture, and intention (Clark, 1996). For example, reactions to a military mannequin will be very different from those to the representation of a parson. Male voices tend to be treated as more authoritative than female voices.

Use of humanlike forms is feasible with prerecorded video and photographs; however, the need depends on the application. Video representation of the lecturer can augment presentations, and video communication helps interactive dialogue. A good speaker holds our attention by a variety of tricks, such as maintaining eye contact, varying the voice tone, and using simple and concise language, as well as delivering an interesting message. These general effects can be reinforced by projected personality. Friendly people are preferred over colder, more hostile individuals. TV announcers, who tend to be middle-aged and confident, but avuncular, characters, have attention drawing power of a dominant yet friendly personality. Both sexes pay attention to extrovert, young personalities, while the male preference for beautiful young women is a particularly strong effect. These traits have been exploited by advertisers for a long time. There are lessons here for multimedia designers as the Web and interactive TV converge. Multimedia will increasingly convey persuasive messages (Reeves & Nass, 1996; Fogg, chap. 17, this Handbook).

Affective effects. Media design for affect (emotional response and arousal) involves both choice of context and interaction. Arousal is increased by (a) interactive applications, (b) surprising events during interaction, (c) use of dynamic media, and (d) challenging images. In contrast, if the objective is to calm the users, arousal can be decreased by choice of natural

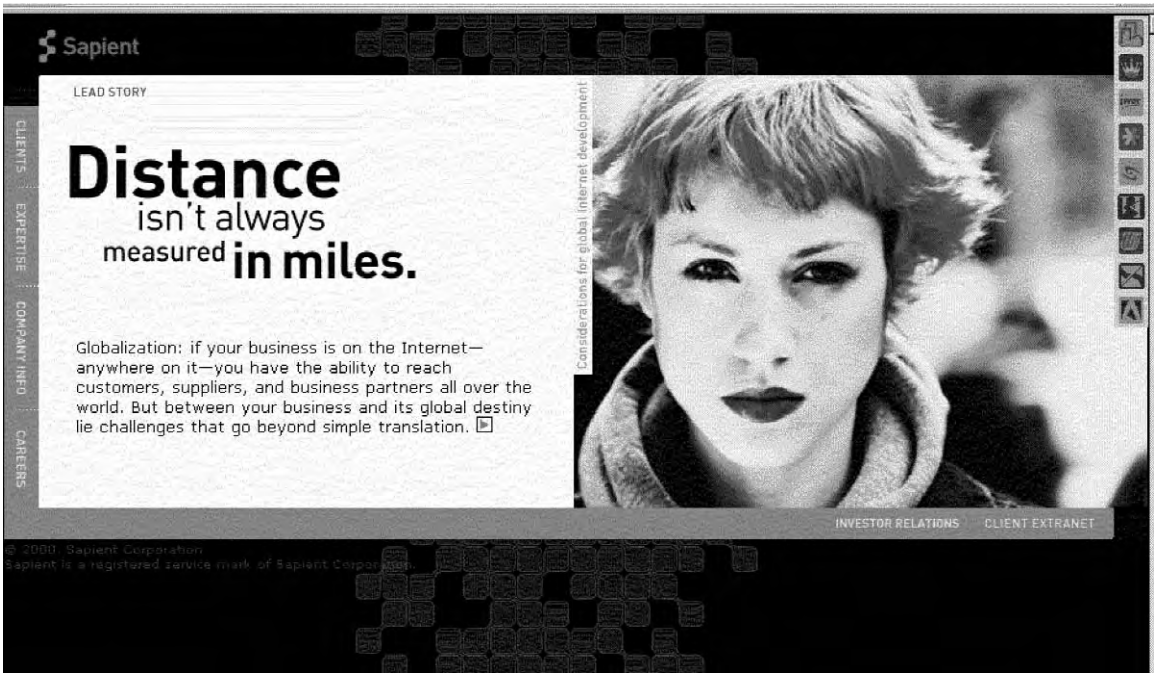


FIGURE 20.6. Effective use of human image for attraction. The picture attracts by the direction of gaze to the user as well as by the appearance of the individual. Courtesy of Sapient.

images and sounds, and soothing music. The most common emotional responses that designers may want to invoke are pleasure, anxiety, fear, and surprise. Pleasure, anxiety, and fear usually depend on our memory of agents, objects, and events (Ortony, Clore, & Collins, 1988), which will be presented, so content selection is the important determinant; however, anxiety can be evoked by uncertainty in interaction and cues to hidden effects. The emotional response of fear or pleasure will depend on the user's previous experience.

Although the bounds of media selection are only set by the creative imagination of the designer, some fundamentals don't change: design still needs to be motivated by users' goals, which may be related to work, learning, or having fun; nevertheless, testing with users is still essential.

The outcome of media selection is a first-cut script. The presentation sequence can be planned using bar charts to illustrate the presentation order, organized according to the task, or application requirements (e.g., lesson plan); see Fig. 20.7. The script gives an outline plan of the presentation, which can be annotated with navigational and dialogue controls. The information model defines the high-level presentation order. Key information items are annotated onto presentation bar charts when planning the sequence and duration of media delivery. The information types are ordered in a first-cut sequence. The selected media are added to the bar chart to show which media stream will be played over time. Decisions on timing depend on the content of the media resource (e.g., length of a video clip, frame display rate). Timeline diagrams can be augmented by hierarchy diagrams to show the classification of information groups that can be mapped to menu style access paths, or network diagrams to help planning hypermedia links.

Interaction and Navigation

Although discussion of interactive multimedia has been delayed until now, in practice, dialogue and presentation design proceed hand in hand. Task analysis provides the basis for dialogue design, as well as specification of navigation controls. Dialogue controls will be included in early storyboards. Navigational and control dialogues allow flexible access to the multimedia content, and give users ability to control how media are played. Dialogue design may also involve specifying how users interact with tools, agents, and objects in interactive microworlds.

Metaphors and interaction design. While task and domain analysis can provide ideas for interaction design, this is also a creative process. Interaction design is essentially a set of choices along a dimension, from simple controls such as menus and buttons, where the user is aware of the interface, to embodiment, in which the user becomes involved as part of the action by controlling an avatar or other representation of their presence. At this end of the dimension multimedia interaction converges with virtual reality (see chap. 31, this Handbook). Interactive metaphors occupy the middle ground.

Some interactive metaphors are generally applicable, such as (a) timelines to move through historical information, (b) the use of a compass to control direction of movement in an inter-

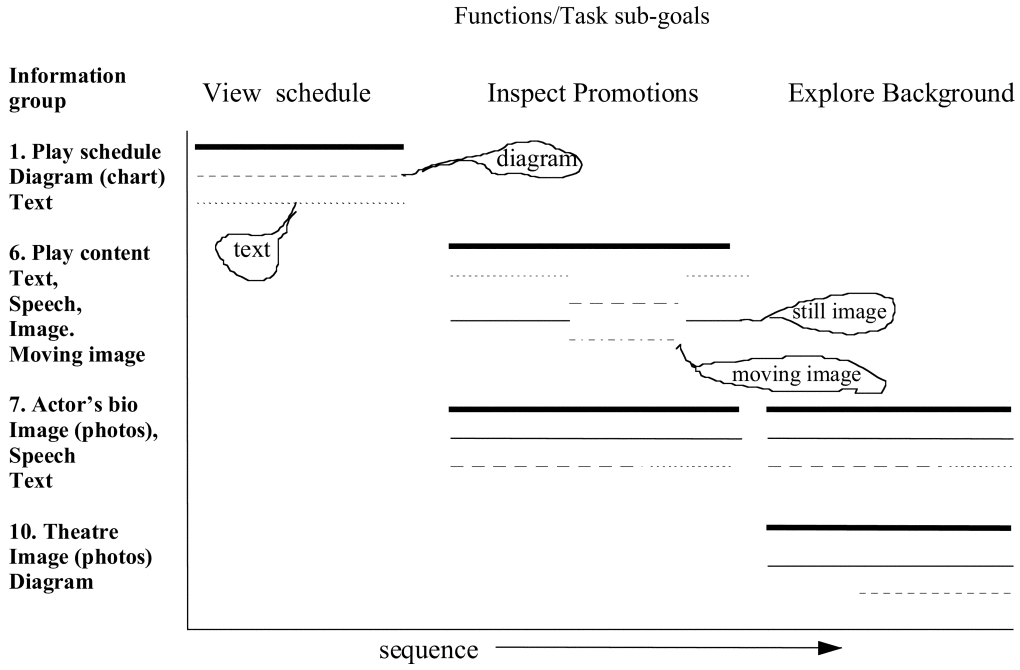


FIGURE 20.7. Bar chart representation of the first-cut presentation script after media selection. Information types are placed on the *y* axis with the task sub-goals that determine the script order on the *x* axis with media resources represented by bars. For this browsing task, three separate sequences are planned for the view schedule, promotions, and so forth.

active space, and (c) controls based on automobiles (steering wheels) or ships (rudders). Others will be more specific, such as selecting and interacting with different characters (young, old, male, female, overweight, fit, etc.) in a health promotion application. Design of interaction also involves creating the microworld within which the user moves, as well as interactive objects that can be selected and manipulated.

Interaction via characters and avatars can increase the user's sense of engagement first by selecting or even constructing the character, although some users may not have the patience to build their own avatar using a graphical paint program. In character-based interaction, the user can either see the world from an egocentric viewpoint (e.g., from their character's position) or exocentric viewpoint (e.g., when they see their character in the graphical world). The sophistication in control of movement and interaction will depend on the hardware available (e.g., joystick, wand, or standard mouse and keyboard). While mimicking physical interaction via datagloves and tracking requires virtual reality technology, relatively complex interaction (e.g., actions in a football game: pass, head ball in direction north/south/east/west) can be programmed into button and function keys. Engagement is also promoted by surprise and unexpected effects, so as the user moves into a particular area a new subworld opens up, or system-controlled avatars appear. These techniques are well known to games programmers; however, they are also applicable to other genres of multimedia applications. The design concepts for engagement can be summarized as follows:

- Character-driven interaction: Provides the user with a choice of avatars or personae they can adopt as representations of themselves within the interactive virtual world.
- Tool-based interaction: Places tools in the world that users can pick up, and the tool becomes the interface (e.g., a virtual mirror magnifies, and a virtual helicopter flies; Tan, Robertson, & Czerwinski, 2001).
- Collaborative characters: In computer mediated communication, these characters may represent other users; in other applications, system-controlled avatars appear to explain, guide, or warn the user.
- Surprise effects: Although conventional HCI guidelines should encourage making the affordances and presence of interactive objects explicit, when designing for engagement, hiding and surprise are important.

Interaction design for an explanatory/tutorial application is illustrated in Fig. 20.8 (www.bbc.co.uk/sn/). This is an interactive microworld in which the user plays the role of a dinosaur character, illustrating use of the engagement concepts. A compass navigation metaphor allows the user to act as the dinosaur, moving around the landscape illustrated in the photographs. The user is given feedback on the characteristics of other predators and prey in the vicinity and has to decide whether to attack or avoid them. Other controls that might be added to such interactive microworlds could be settings to change the environment (e.g., add more predators, change the weather, etc.). Engagement can be taken even further by giving the user facilities to actually design the MacWorld so the application becomes a domain-oriented design environment (Fischer, Giaccardi, Ye, Sutcliffe, & Mehandjiev, 2004).

Navigation. In information-intensive multimedia where access to content is the main design goal, hypermedia dialogues

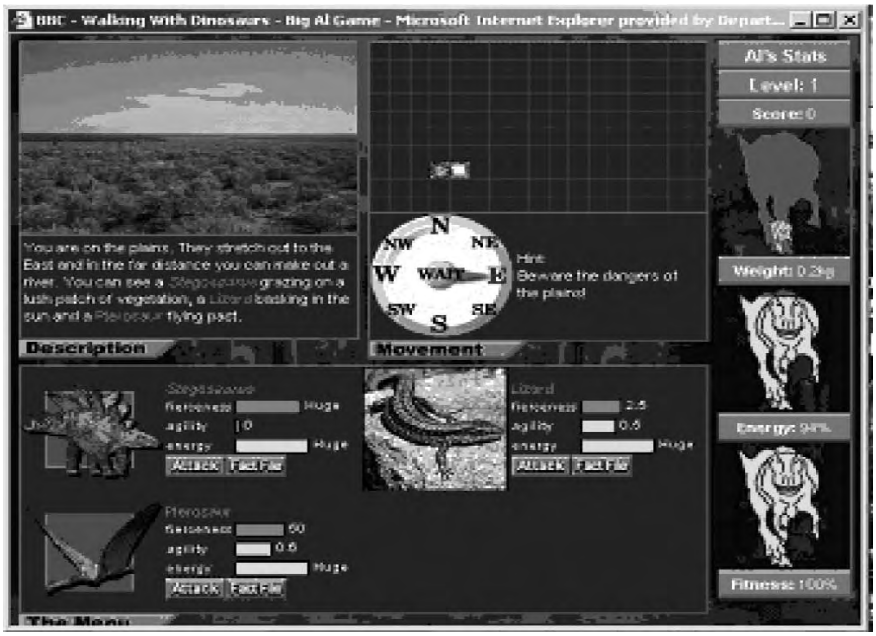


FIGURE 20.8. Interactive microworld—Big Al game (www.bbc.co.uk/sn/). The user plays the dinosaur role by navigating with the compass metaphor. The photograph updates with each move and the user is given a choice of attacking or avoiding other dinosaurs in the virtual world.

that link content segments will be appropriate. Good hypertext design is based on a sound information analysis that specifies the pathways between related items, and the use of cues to show the structure of the information space to the user. In document-based hypermedia (e.g., HTML and the Web), links can only access the whole media resource rather than point to components within it. The design issues are to (a) plan the overall structure, (b) segment complex structures into a hierarchy of subnetworks, and then (c) plan the implementation of each subnetwork. The access structure of most hypermedia will be hierarchical, organized according to the information model and categorization of content (e.g., information grouped by function, organization, task usage, or user preference). Navigation design transforms the user's conceptual model of an information space into a hypermedia structure. Unfortunately, individual users have different models, so this may not be an easy task. Implementing too many links in order to satisfy each user's view will make the system too complex and increase the chance of the user getting lost. Too few links will frustrate users who cannot find the associations they want. Unfortunately, hypermedia systems assume a fixed link structure so the user is limited to the pathways provided by the designer. More open-ended hypermedia environments (e.g., Microcosm: Lowe & Hall, 1998) provide more flexibility via links with query facilities which allow access to databases. Dynamic links attached to hotspots in images or nodes in text documents provide access paths to a wider variety of data.

One problem with large hypermedia systems is that users get lost in them. Navigation cues, waymarks, and mini-map overviews can help to counter the effects of disorientation. *Mini-maps* give an overview of the hypertext area and a reference context for where users are in large networks. *Filters* help to reduce complexity by showing only a subset of nodes and links that the user is interested in. Having typed links helps filtering views because the user can guess the information content from the link type (e.g., reference, example, source, related work, etc.). Other navigation facilities are *visit lists* containing a history of nodes traversed in a session, and *bookmarks,* so users can tailor a hypermedia application with their own navigation aide-memoires (Nielsen, 1995). Once the structure has been designed, link cues need to be located within media resources, so the appropriate cues need to be considered for each medium. For example,

- *Text media:* The Web convention is to underline and highlight text in a consistent color, such as blue or purple.
- *Images:* Link cues can be set as standalone icons or as active components in images. Icons need to be tested with users because the designer's assumed meaning can be ambiguous. Active components should signal the link's presence by captions or pop-up hover text so the user can inspect a link before deciding whether to follow it.
- *Moving images:* Links from animation and film are difficult to design because the medium is dynamic; however, link buttons can be placed below the video window. Active components, such as overlaid buttons within a moving image, are technically more challenging programs. Buttons may also be timed to pop up at appropriate times during the video.
- *Audio links:* Sound and speech links present the same difficulties as moving images. One solution is to use visual cues, possibly synchronized with the sound or speech track. If speech recognition is available, then voice commands can act as links, but these commands need to be explained to the user.

Navigation controls provide access to the logical content of multimedia resources; however, access to logical components may be constrained by limitations of physical media resources. For example, a movie may be logically composed of several scenes, but it can only be accessed by an approximate timer set to the beginning of the whole film. Worse still, a video clip may only be playable as a single segment, making implementation of navigation requirements impossible. It is advisable to give the user control whenever possible.

In many cases, the media-rendering device, such as a video player for .avi files or QuickTime movies, will provide controls. If controls have to be implemented from scratch, the following list should be considered for each media type:

- *Static media:* Size and scale controls to zoom and pan; page access, if the medium has page segmentation, as in text and diagrams; the ability to change attributes such a color and display resolution; and font type and size in text.
- *Dynamic media:* The familiar video controls of stop, start, play, pause, fast forward and rewind, as well as the ability to address a particular point or event in the media stream by a time marker or an index (e.g., "go to" component/marker, etc.).

Navigation controls employ standard user interface components (e.g., buttons, dialogue boxes, menus, icons, and sliders) and techniques (e.g., form filling, dialogue boxes, and selection menus). For more guidance, see ISO 9241, parts 12, 14 & 17 (International Standards Organization, 1997) and ISO 14915, part 2 (International Standards Organization, 1998).

Design for Attention

Having selected the media resources, the designer must now ensure that the user will extract the appropriate information. An important consideration of multimedia design is to link the thread of a message across several different media. This section gives recommendations on planning the user's reading/viewing sequence, and guidelines for realizing these recommendations in presentation sequences, hypermedia dialogues, and navigation controls. The essential differences are timing and user control. In a presentation design, the reading/viewing sequence and timing are set by the designer, whereas the reading/viewing sequence in hypertext implementation and interactive dialogues is under user control.

Presentation techniques help to direct the user's attention to important information and specify the desired order of reading or viewing. The need for thematic links between information components is specified, and attention-directing techniques are selected to implement the desired effect.

The design issues are:

- To plan the overall thematic thread of the message.
- To draw the user's attention to important information.
- To establish a clear reading/viewing sequence.

- To provide clear links when the theme crosses from one medium to another.

Design for attention is particularly important for images. User attention to time-varying media is determined by the medium itself (e.g., we have little choice but to listen to speech or to view animations in the order in which they are presented). The reading sequence is directed by the layout of text, although this is culturally dependent (e.g., western languages read left to right, Arabic languages in the opposite direction). Viewing order in images, however, is unpredictable unless the design specifically selects the user's attention.

In some cases, a common topic may be sufficient for directing the user's reading/viewing sequence; however, when the thread is important or hard to follow, designed effects for attention, or contact points, are necessary to aid the user's perception. The term *contact point* refers to a reference from one medium to another and comes from the research of Hegarty and Just (1993) and Narayanan and Hegarty (1998), which demonstrated that comprehension is improved by reinforcing the links between information in different media. Two types of contact point are distinguished:

1. *Direct contact points:* Attention-directing effects are implemented in both the source and destination medium (e.g., in the text, an instruction such as, "Look at the oblong component in Figure 1" is given while the component is highlighted). Direct contact points create a strong cross reference between two media, but can become intrusive if overused. An example of a direct contact point is shown in Fig. 20.9 (Faraday & Sutcliffe, 1997a).
2. *Indirect contact points* implement an attention-directing effect only in the source, or less frequently the destination, medium (e.g., "In Figure 1, the assembly is shown" is spoken, with no highlighting being used in the image.). Indirect contact points have less attention-directing force and work by temporal sequencing or spatial juxtapositoning. Indirect contact points are less intrusive, so they may be used more frequently without becoming disruptive.

In most cases, the attentional effect in a direct contact point will be actuated in sequence, although occasionally, both effects may be presented concurrently if the order of the association is not important. In hypermedia implementations, direct contact points become a link cue in the source medium and a highlight anchor in the destination medium.

Multiple contact points may be organized in a logical order to follow the theme and connect a thread of topics (Faraday & Sutcliffe, 1998a, 1999). For instance, in a biology tutorial, the explanation of parts of a cell is organized with interleaved

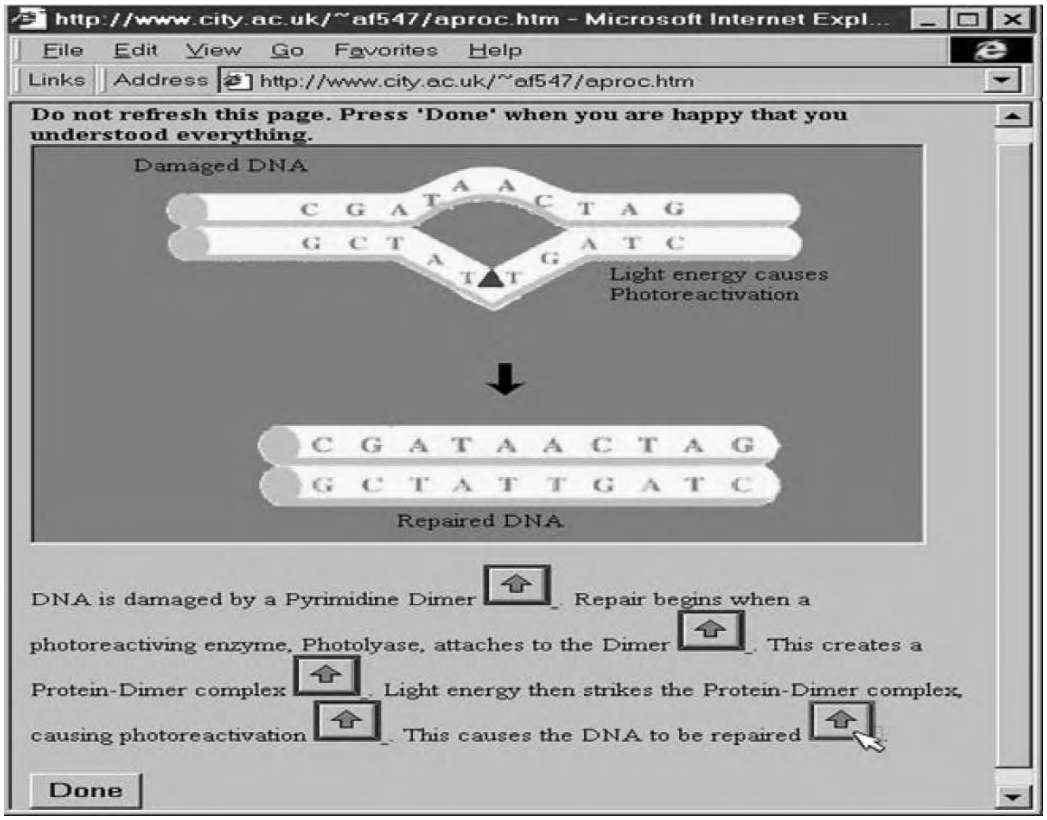


FIGURE 20.9. Direct contact point between text and video segments, reinforced by speech. The arrows play video clips linked to the text description. The user can either play the whole explanation or step through it using the contact point links (Faraday & Sutcliffe, 1997a).

speech segments and a diagram describing the cell's components top to bottom, left to right. Highlighting techniques locate each component in turn, following the order of the spoken explanation. The attention-directing techniques described in the following section are used to implement contact points.

The design problem is how to direct the user's attention to the appropriate information at the correct level of detail. Initially, users will tend to extract information from images at the scene level, (e.g, major objects will be identified but with very little descriptive detail; Treisman, 1988). Regular layout grids help design composite images (Mullet & Sano, 1995) and encourage viewing sequences in image sets. Alternatively, the window frame can be set to control which parts of an image are viewed. Larger window frames will be attended to before smaller areas. A list of the key components that the user needs to focus on and the facts that should be extracted is checked against the initial presentation design to see if the key components will attract sufficient attention, or whether the user is likely to be confused by extraneous detail.

Still image media. Highlighting techniques for designed and natural images, organized in approximate power of their effect, are summarized in Table 20.2. A common highlighting technique will pick out spatially distributed objects, such as when all related objects are changed to the same color; colocated objects can be grouped by using a common color or texture for their background, or by drawing a box around them. The highlighted area will set the granularity of the user's attention. Captions linked to objects in an image are another useful means of drawing attention and providing supplementary information (e.g., identity). Dynamic revealing of captions is particularly effective for directing the user's viewing sequence. Sequential highlighting is also useful for showing the pathways or navigational instructions.

Moving image media. Directing attention to components within moving images is difficult because of the dynamic nature of the medium. Design of film and video is an extensive subject in its own right, so treatment here will necessarily be brief. The following design advice is based on Hochberg (1986). The design objectives, as for other media, are to draw the user's attention to key components within the video or animation.

First, the content should be structured into scenes that correspond to the information script. To structure animation sequences and make scene boundaries obvious, use a cut, wipe, or dissolve to emphasize that a change in the content structure has taken place; however, cuts should be used with care and continuity maintained between the two sequences if they are to be integrated. *Continuity* is manifest as the same viewpoint and subject matter in two contiguous shots. Change in background or action (e.g., an individual walking left in one clip and walking right in the next) is quickly noticed as a change. An establishing shot that shows the whole scene helps to introduce a new sequence and provide context. To provide detail of a newly introduced object or context, the object is shown filling the frame with a small amount of surrounding scene, while to imply a relationship or compare two objects, a tight two-shot with both objects together in the same frame is advisable.

Linguistic media (text and speech). As with moving image, the literature is extensive, so the following heuristics are a brief summary; see Levie and Lentz (1982) for more detail. Text may be structured to indicate subsections by indentation, formatting into paragraphs or columns, or segmented by background color. Bullet points or numbered sections indicate order more formally, such as for procedures. Different voices help to structure speech while also attracting attention. If language is being used to set the context for accompanying media, it is important that the correct level of identification is set. For instance, a higher level concept, or the whole scene in an accompanying image, is described at the beginning of a script, and then lower level topics reset the user's focus. Discourse markers can make phrases and sentences more salient.

Adding contact points and attention-directing effects completes the design process; however, as with all user interfaces, there is no substitute for usability testing. Designs are con-

TABLE 20.2. Attention-Directing Techniques for Different Media

	Attention Orienting Techniques in Approximate Order of Power	Notes
Still image: designed and natural	Movement of or change in the shape/size/color of an object. Use of bold outline. Object marked with a symbol (e.g., arrow) or icon. Draw boundary, use color, shape, size, or texture to distinguish important objects.	Some effects may compromise natural images because they overlay the background image with new components (e.g., arrows, arcs, icons). Group objects by a common technique.
Moving image	Freeze frame followed by applying a still image highlight. Zoom, close-up shot of the object. Cuts, wipes, and dissolve effects.	Change in topographic motion, in which an object moves across the ground of an image, is more effective than internal movement of an object's components. Size and shape may be less effective for highlighting a moving object.
Text	Bold, font size, type, color, or underlining. To direct attention to larger segments of text, use formatting, bullet points, subsections, and/or indentation.	Formatting techniques are paragraphs; headings/titles as entry points; and indents to show hierarchical nesting, with bullet points and lists.
Speech/sound	Familiar voice. Silence followed by onset of sound. Different voices, or a change in voice prosody (tonality), amplitude (loudness), change and variations in pitch (frequency), voice rate, change source direction, and alarm sounds (police sirens).	Voices familiar to the user (e.g., close relatives) attract attention over nonfamiliar speech. Discourse markers such as "Next," "Because," "So," and so forth draw attention to subsequent phrases.

structed incrementally by iterations of design and evaluation that checks for usability using standard methods, with additional memory and comprehension tests for multimedia. Therefore, when testing a design, ask the user to tell you what they understood the message to be. This can be done during the presentation with a think-aloud protocol to check that users did attend to key items, and afterwards, by a memory test. If a user cannot remember key components in the message, then the design may need to be improved.

CONCLUSIONS

Multimedia still poses many issues for further research. The design method described in this chapter, coupled with user-centred design, can improve quality; however, there is still a need for experts to create specific media resources (e.g., film/video and audio experts). Furthermore, considerable research is still necessary before we fully understand the psychology of multimedia interaction. Design for motivation and attractiveness is still poorly understood, and personality effects in media may not be robust when usability errors impede communication. The process by which people extract information from complex images still requires extensive research, although the increasing number of eyetracking studies is beginning to throw some light on this topic. In the future, language and multimodal communication will change our conception of multimedia from its current CD-ROM or Web-based form into conversational and multisensory interfaces. Multimedia will become part of wearable and ubiquitous user interfaces, where the media are part of our everyday environment. Design for multisensory communication will treat media and artifacts (e.g., our desks, clothing, walls in our homes) as a continuum, while managing the diverse inputs to multimedia from creative design, technology, and usability engineering will be one of the many interesting future challenges.

References

Alty, J. L. (1991). Multimedia: what is it and how do we exploit it? In D. Diaper & N. V. Hammond (Eds.), *Proceedings of HCI '91: People and Computers VI* (pp. 31–41). Cambridge, U.K.: Cambridge University Press.

Alty, J. L. (1997). Multimedia. In A. B. Tucker (Ed.), *Computer science and engineering handbook* (pp. 1551–1570). New York: CRC Press.

Arens, Y., Hovy, E., & Van Mulken, S. (1993). Structure and rules in automated multimedia presentation planning. *Proceedings: IJCAI-93: Thirteenth International Joint Conference on Artificial Intelligence*. Chambéry, France (pp. 231–235). San Mateo, CA: Morgan-Kaufmann

Ayers P., & Sweller, J. (2005). The split attention principle for multimedia learning. In R. E. Mayer (Ed.), *The Cambridge Handbook for Multimedia Learning* (pp 135–146). Cambridge, U.K.: Cambridge University Press

Baddeley, A. D. (1986). *Working memory*. Oxford, U.K.: Oxford University Press.

Barnard, P. (1985). Interacting cognitive subsystems: A psycholinguistic approach to short term memory. In A. Ellis (Ed.), *Progress in psychology of language: Vol. 2.* (pp. 197–258). London: Lawrence Erlbaum Associates.

Bernsen, N. O. (1994). Foundations of multimodal representations: a taxonomy of representational modalities. *Interacting with Computers, 6*(4), 347–371.

Bertin, J. (1983). *Semiology of graphics*. Madison, WI: University of Wisconsin Press.

Bieger, G. R., & Glock, M. D. (1984). The information content of picture-text instructions. *Journal of Experimental Education, 53*, 68–76.

Booher, H. R. (1975). Relative comprehensibility of pictorial information and printed word in proceduralized instructions. *Human Factors, 17*(3), 266–277.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Clark, H. H. (1996). *Using language*. Cambridge, U.K.: Cambridge University Press.

Faraday, P., & Sutcliffe, A. G. (1996). An empirical study of attending and comprehending multimedia presentations. *Proceedings ACM Multimedia 96: 4th Multimedia Conference,* Boston (pp. 265–275). New York: ACM Press.

Faraday, P., & Sutcliffe, A. G. (1997a). Designing effective multimedia presentations. *Proceedings: Human Factors in Computing Systems CHI'97,* Atlanta, GA (pp. 272–279). New York: ACM Press.

Faraday, P., & Sutcliffe, A. G. (1997b). Multimedia: design for the moment. *Proceedings: Fifth ACM International Multimedia Conference*, Seattle, WA (pp. 183–192). New York: ACM Press.

Faraday, P., & Sutcliffe, A. G. (1998a). Providing advice for multimedia designers. In C. M. Karat, A. Lund, J. Coutaz, & J. Karat, (Eds.), *CHI 98 Conference Proceedings: Human Factors in Computing Systems,* Los Angeles (pp. 124–131). New York: ACM Press.

Faraday, P., & Sutcliffe, A. G. (1998b). Using contact points for Web page design. *People and Computers XIII, Proceedings: BCS-HCI Conference,* Sheffield (pp. 20–22). Berlin: Springer-Verlag.

Faraday, P., & Sutcliffe, A. G. (1999). Authoring animated Web pages using contact points. *Proceedings of CHI'99: Human Factors in Computing Systems,* Pittsburgh, PA (pp. 458–465). New York: ACM Press.

Fischer G., Giaccardi E., Ye, Y.,Sutclifffe A.G., & Mehandjiev, N. (2004), Meta-Design: A manifesto for End User development. *Communications of the ACM, 47*(9), pp. 33–37.

Fogg, B. J. (1998). Persuasive computer: perspectives and research directions. *Proceedings: Human Factors in Computing Systems: CHI '98,* Los Angeles (pp. 225–232). New York: ACM Press.

Gardiner, M., & Christie, B. (1987). *Applying cognitive psychology to user interface design*. Chichester, NY: Wiley.

Hegarty, M., & Just, M. A. (1993). Constructing mental models of text and diagrams. *Journal of Memory and Language, 32*, 717–742.

Heller, R. S., & Martin, C. (1995). A media taxonomy. *IEEE Multimedia* (Winter), 36–45.

Hochberg, J. (1986). Presentation of motion and space in video and cinematic displays. In K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of perception and human performance, 1: Sensory processes and perception*. New York: Wiley.

Hollan, J. D., Hutchins, E. L., & Weitzman, L. (1984). Steamer: an interactive inspectable simulation-based training system. *AI Magazine, 5*(2), 15–27.

International Standards Organization. (1997). *ISO 9241: Ergonomic requirements for office systems with visual display terminals (VDTs)*. Geneva: ISO.

International Standards Organization. (1998). *ISO 14915 Multimedia user interface design software ergonomic requirements, Part 1: Introduction and framework; Part 3: Media combination and selection*. Geneva: ISO.

International Standards Organization. (2000). *ISO 14915-3: Software ergonomics for multimedia user interfaces. Part 3: Media selection and combination. Draft international standard*. Geneva: ISO.

Kristof, R., & Satran, A. (1995). *Interactivity by design: creating and communicating with new media*. Mountain View, CA: Adobe Press.

Lavie, T., & Tractinsky, N. (2004). Assessing dimensions of perceived visual aesthetics of websites. *International Journal of Human-Computer Studies, 60*(3), 269–298.

Levie, W. H., & Lentz, R. (1982). Effects of text illustrations: a review of research. *Educational Computing and Technology Journal, 30*(4), 159–232.

Lowe, D., & Hall, W. (1998). *Hypermedia and the Web*. Chichester, NY: Wiley.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical Structure Theory: toward a functional theory of text organisation. *Text, 8*(3), 243–281.

May, J., & Barnard, P. (1995). Cinematography and interface design. In K. Nordbyn, P. H. Helmersen, D. J. Gilmore, & S. A. Arnesen, (Eds.), *Proceedings: Fifth IFIP TC 13 International Conference on Human-Computer Interaction,* Lillehammer, Norway (pp. 26–31). London: Chapman & Hall.

Mullet, K., & Sano, D. (1995). *Designing visual interfaces: communication oriented techniques*. Englewood Cliffs, NJ: SunSoft Press.

Narayanan, N. H., & Hegarty, M. (1998). On designing comprehensible interactive hypermedia manuals. *International Journal of Human-Computer Studies, 48*, 267–301.

Narayanan, N. H., & Hegarty, M. (2002). Multimedia design for communication of dynamic information. *International Journal of Human-Computer Studies, 57*(4), 279–316.

Nielsen, J. (1995). *Multimedia and hypertext: the internet and beyond*. Boston: AP Professional.

Norman, D. A. (2004). *Emotional design: Why we love (or hate) everyday things*. New York: Basic Books.

Norman, D. A., & Shallice, T. (1986). Attention to action: willed and automatic control of behaviour. In G. E. Davidson, & G. E. Schwartz (Eds.), *Consciousness and self-regulation, Vol. 4* (pp. 1–18). New York: Plenum.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions*. Cambridge, U.K.: Cambridge University Press.

Oviatt S., Darves C., & Coulston R. (2004). Towards adaptive conversational interfaces: Modelling speech convergence with animated personas. *ACM Transactions of Computer Human Interaction, 11*(3), 300–328.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Reeves, B., & Nass, C. (1996). *The media equation: how people treat computers, television and new media like real people and places*. Stanford, CA/Cambridge: CLSI/Cambridge University Press.

Rogers, Y., & Scaife, M. (1998). How can interactive multimedia facilitate learning? In J. Lee (Ed.), *Intelligence and multimodality in multimedia interfaces: research and applications*. Menlo Park, CA: AAAI Press.

Scaife, M., Rogers, Y., Aldrich, F., & Davies, M. (1997). Designing for or designing with? Informant design for interactive learning environments. In Pemberton S. (Ed.), *Proceedings: Human Factors in Computing Systems CHI'97*, Atlanta, GA (pp. 343–350). New York: ACM Press.

Spool, J. M., Scanlon, T., Snyder, C., Schroeder, W., & de Angelo, T. (1999). *Web site usability: a designer's guide*. San Francisco: Morgan Kaufmann.

Sutcliffe, A. G. (1997). Task-related information analysis. *International Journal of Human-Computer Studies, 47*(2), 223–257.

Sutcliffe, A. G. (1999). User-centred design for multimedia applications. *Proceedings Vol. 1: IEEE Conference on Multimedia Computing and Systems,* Florence, Italy (pp. 116–123). Los Alamitos, CA: IEEE Computer Society Press.

Sutcliffe, A. G. (2002). Assessing the reliability of heuristic evaluation for website attractiveness and usability. *Proceedings HICSS-35: Hawaii International Conference on System Sciences,* Honolulu (pp. 1838–1847). Los Alamitos, CA: IEEE Computer Society Press.

Sutcliffe, A. G. (2003). *Multimedia and virtual reality: designing multisensory user interfaces*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sutcliffe, A. G., & Faraday, P. (1994). Designing presentation in multimedia interfaces. In B. Adelson, S. Dumais, & J. Olson, (Eds.), *CHI '94 Conference Proceedings: Human Factors in Computing Systems 'Celebrating Interdependence',* Boston (pp. 92–98). New York: ACM Press.

Sutcliffe, A. G., & de Angeli, A. (2005). Assessing interaction styles in web user interfaces. In M. F. Costabile & F. Paterno (Eds.), *Proceedings of Human computer Interaction- INTERACT 05,* Rome (pp. 405–417) Berlin: Springer-Verlag.

Sutcliffe, A. G., Kurniawan, S., & Shin, J-E. (2006). A Method and advisor tool for multimedia user interface design. *International Journal of Human Computer Studies*, *64*(4), 375–312.

Tan, D. S., Robertson, G. R., & Czerwinski, M. (2001). Exploring 3D navigation: Combining speed coupled flying with orbiting. In J. A. Jacko, A. Sears, M. Beaudouin-Lafon, & R. J. K. Jacob, (Eds.), *CHI 2001 Conference Proceedings: Conference on Human Factors in Computing Systems,* Seattle, WA, (pp. 418–425). New York: ACM Press.

Tractinsky, N. (1997). Aesthetics and apparent usability: Empirically assessing cultural and methodological issues. In Pemberton S. (Ed.), *Human Factors in Computing Systems: CHI 97 Conference Proceedings,* Atlanta, GA (pp. 115–122). New York: ACM Press.

Tractinsky, N., Shoval-Katz, A., & Ikar, D. (2000). What is beautiful is usable. *Interacting with Computers, 13*(2), 127–145.

Treisman, A. (1988). Features and objects: fourteenth Bartlett memorial lecture. *Quarterly Journal of Experimental Psychology, 40A*(2), 201–237.

Tufte, E. R. (1997). *Visual explanations: images and quantities, evidence and narrative*. Cheshire, CT: Graphics Press.

Wickens, C. D., Sandry, D., & Vidulich, M. (1983). Compatibility and resource competition between modalities of input, output and central processing. *Human Factors, 25*, 227–248.

21

MULTIMODAL INTERFACES

Sharon Oviatt
Oregon Health & Science University

What are Multimodal Systems, and Why Are We Building Them? **414**
What Types of Multimodal Interfaces Exist, and What Is Their History and Current Status? **414**
What Are the Goals and Advantages of Multimodal Interface Design? **417**
What Methods and Information Have Been Used to Design Novel Multimodal Interfaces? **420**
What Are the Cognitive Science Underpinnings of Multimodal Interface Design? **421**
When Do Users Interact Multimodally? 421
What Are the Integration and Synchronization Characteristics of Users' Multimodal Input? 422
What Individual Differences Exist in Multimodal Interaction, and What Are the Implications for Designing Systems for Universal Access? 423
Is Complementarity or Redundancy the Main Organizational Theme That Guides Multimodal Integration? 424
What Are the Primary Features of Multimodal Language? 424
What Are the Basic Ways in Which Multimodal Interfaces Differ From Graphical User Interfaces? **425**
What Basic Architectures and Processing Techniques Have Been Used to Design Multimodal Systems? **425**
What Are the Main Future Directions for Multimodal Interface Design? **427**
Acknowledgments **428**
References **428**

WHAT ARE MULTIMODAL SYSTEMS, AND WHY ARE WE BUILDING THEM?

Multimodal systems process two or more combined user input modes—such as speech, pen, touch, manual gestures, gaze, and head and body movements—in a coordinated manner with multimedia system output. This class of systems represents a new direction for computing, and a paradigm shift away from conventional WIMP interfaces. Since the appearance of Bolt's (1980) "Put That There" demonstration system, which processed speech in parallel with touch-pad pointing, a variety of new multimodal systems has emerged. This new class of interfaces aims to recognize naturally occurring forms of human language and behavior, which incorporate at least one recognition-based technology (e.g., speech, pen, vision). The development of novel multimodal systems has been enabled by the myriad input and output technologies currently becoming available, including new devices and improvements in recognition-based technologies. This chapter will review the main types of multimodal interfaces, their advantages and cognitive science underpinnings, primary features and architectural characteristics, and general research in the field of multimodal interaction and interface design.

The growing interest in multimodal interface design is inspired largely by the goal of supporting more transparent, flexible, efficient, and powerfully expressive means of human-computer interaction. Multimodal interfaces are expected to be easier to learn and use, and are preferred by users for many applications. They have the potential to expand computing to more challenging applications, to be used by a broader spectrum of everyday people, and to accommodate more adverse usage conditions than in the past. Such systems also have the potential to function in a more robust and stable manner than unimodal recognition systems involving a single recognition-based technology, such as speech, pen, or vision.

The advent of multimodal interfaces based on recognition of human speech, gazes, gestures, and other natural behaviors represents only the beginning of a progression toward computational interfaces capable of relatively humanlike sensory perception. Such interfaces eventually will interpret continuous input from a large number of different visual, auditory, and tactile input modes, which will be recognized as users engage in everyday activities. The same system will track and incorporate information from multiple sensors on the user's interface and surrounding physical environment in order to support intelligent adaptation to the user, task, and usage environment. Future adaptive multimodal-multisensor interfaces have the potential to support new functionality, to achieve unparalleled robustness, and to perform flexibly as a multifunctional and personalized mobile system.

WHAT TYPES OF MULTIMODAL INTERFACES EXIST, AND WHAT IS THEIR HISTORY AND CURRENT STATUS?

Multimodal systems have developed rapidly during the past decade, with steady progress toward building more general and robust systems, as well as more transparent human interfaces than ever before (Benoit, Martin, Pelachaud, Schomaker, & Suhm, 2000; Oviatt et al., 2000). Major developments have occurred in the hardware and software needed to support key component technologies incorporated within multimodal systems, as well as in techniques for integrating parallel input streams. Multimodal systems also have diversified to include new modality combinations, including speech and pen input, speech and lip movements, speech and manual gesturing, and gaze tracking and manual input (Benoit & Le Goff, 1998; Cohen et al., 1997; Stork & Hennecke, 1995; Turk & Robertson, 2000; Zhai, Morimoto, & Ihde, 1999). In addition, the array of multimodal applications has expanded extremely rapidly in recent years. Among other areas, it presently includes multimodal map-based systems for mobile and in-vehicle use; multimodal browsers; multimodal interfaces to virtual reality systems for simulation and training; multimodal person-identification/verification systems for security purposes; multimodal medical, educational, military, and web-based transaction systems; and multimodal access and management of personal information on handhelds and cell phones (Cohen & McGee, 2004; Iyengar, Nock, & Neti, 2003; McGee, 2003; Neti, Iyengar, Potamianos, & Senior, 2000; Oviatt, 2003; Oviatt, Flickner, & Darrell, 2004; Oviatt & Lunsford, 2005; Oviatt et al., 2000; Pankanti, Bolle, & Jain, 2000; Reithinger et al., 2003).

In one of the earliest multimodal concept demonstrations, Bolt had users sit in front of a projection of "Dataland" in "the Media Room" (Negroponte, 1978). Using the "Put That There" interface (Bolt, 1980), they could use speech and pointing on an armrest-mounted touchpad to create and move objects on a 2D large-screen display. For example, the user could issue a command, such as "Create a blue square there," with the intended location of "there" indicated by a 2D cursor mark on the screen. Semantic processing was based on the user's spoken input, and the meaning of the deictic "there" was resolved by processing the x/y coordinate indicated by the cursor at the time "there" was uttered. Since Bolt's early prototype, considerable strides have been made in developing a wide variety of different types of multimodal systems.

Among the earliest and most rudimentary multimodal systems were ones that supported speech input along with a standard keyboard and mouse interface. Conceptually, these multimodal interfaces represented the least departure from traditional graphical user interfaces (GUIs). Their initial focus was on providing richer natural language processing to support greater expressive power for the user when manipulating complex visuals and engaging in information extraction. As speech recognition technology matured during the late 1980s and 1990s, these systems added spoken input as an alternative to text entry via the keyboard. As such, they represent early involvement of the natural language and speech communities in developing the technologies needed to support new multimodal interfaces. Among the many examples of this type of multimodal interface are CUBRICON, Georal, Galaxy, XTRA, Shoptalk, and Miltalk (Cohen et al., 1989; Kobsa et al., 1986; Neal & Shapiro, 1991; Seneff, Goddeau, Pao, & Polifroni, 1996; Siroux, Guyomard, Multon, & Remondeau, 1995; Wahlster, 1991).

Several of these early systems were multimodal-multimedia map systems to which a user could speak or type and point with a mouse to extract tourist information or engage in military sit-

uation assessment (Cohen et al., 1989; Neal & Shapiro, 1991; Seneff et al., 1996; Siroux et al., 1995). For example, using the CUBRICON system a user could point to an object on a map and ask, "Is this <point> an air base?" CUBRICON was an expert system with extensive domain knowledge, as well as natural language processing capabilities that included referent identification and dialogue tracking (Neal & Shapiro, 1991). With the Georal system, a user could query a tourist-information system to plan travel routes using spoken input and pointing via a touch-sensitive screen (Siroux et al., 1995). In contrast, the Shoptalk system permitted users to interact with complex graphics representing factory production flow for chip manufacturing (Cohen et al., 1989). Using Shoptalk, a user could point to a specific machine in the production layout and issue the command: "Show me all the times when this machine was down." After the system delivered its answer as a list of time ranges, the user could click on one to ask the follow-up question: "What chips were waiting in its queue then, and were any of them hot lots?" Multimedia system feedback was available in the form of a text answer, or the user could click on the machine in question to view an exploded diagram of the machine queue's contents during that time interval.

More recent multimodal systems have moved away from processing simple mouse or touchpad pointing, and have begun designing systems based on two parallel input streams that each are capable of conveying rich semantic information. These multimodal systems recognize two natural forms of human language and behavior, for which two recognition-based technologies are incorporated within a more powerful bimodal user interface. To date, systems that combine either speech and pen input (Oviatt & Cohen, 2000) or speech and lip movements (Benoit et al., 2000; Stork & Hennecke, 1995; Rubin, Vatikiotis-Bateson, & Benoit, 1998; Potamianos, Neti, Gravier, & Garg, 2003) constitute the two most mature areas within the field of multimodal research. In these cases, the keyboard and mouse have been abandoned. For speech and pen systems, spoken language sometimes is processed along with complex pen-based gestural input involving hundreds of different symbolic interpretations beyond pointing[1] (Oviatt et al., 2000). For speech and lip movement systems, spoken language is processed along with corresponding human lip movements during the natural audio-visual experience of spoken interaction. In both of these sub-literatures, considerable work has been directed toward quantitative modeling of the integration and synchronization characteristics of the two rich input modes being processed, and innovative time-sensitive architectures have been developed to process these patterns in a robust manner.

Multimodal systems that recognize speech and pen-based gestures first were designed and studied in the early 1990s (Oviatt, Cohen, Fong, & Frank, 1992), with the original QuickSet system prototype built in 1994. The QuickSet system is an agent-based collaborative multimodal system that runs on a hand-held PC (Cohen et al., 1997). With QuickSet, for example, a user can issue a multimodal command such as "Airstrips . . . facing this way <draws arrow>, and facing this way <draws arrow>," using combined speech and pen input to place the correct number, length, and orientation (e.g., SW, NE) of aircraft landing strips on a map. Other research-level systems of this type were built in the late 1990s. Examples include the Human-centric Word Processor, Portable Voice Assistant, QuickDoc and MVIEWS (Bers, Miller, & Makhoul, 1998; Cheyer, 1998; Oviatt et al., 2000; Waibel, Suhm, Vo, & Yang, 1997). These systems represent a variety of different system features, applications, information fusion, and linguistic processing techniques. For illustration purposes, a comparison of five different speech and gesture systems is summarized in Fig. 21.1. In most cases, these multimodal systems jointly interpreted speech and pen input based on a frame-based method of information fusion and a late semantic fusion approach, although QuickSet used a statistically-ranked unification process and a hybrid symbolic/statistical architecture (Wu, Oviatt, & Cohen, 1999). Other recent systems also have adopted unification-based multimodal fusion and a hybrid architectural approach for processing multimodal input (Bangalore & Johnston, 2000; Denecke & Yang, 2000; Pfleger, 2004; Wahlster, 2001) and even multimodal output (Kopp, Tepper, & Cassell, 2004).

Multimodal systems that process speech and continuous 3D manual gesturing are emerging rapidly, although these systems remain less mature than ones that process 2D pen input (Encarnacao & Hettinger, 2003; Flanagan & Huang, 2003; Sharma, Pavlovic, & Huang, 1998; Pavlovic, Sharma, & Huang, 1997). This primarily is because of the significant challenges associated with segmenting and interpreting continuous manual movements, compared with a stream of *x/y* ink coordinates. Because of this difference, multimodal speech and pen systems have advanced more rapidly in their architectures, and have progressed further toward commercialization of applications. However, a significant cognitive science literature is available for guiding the design of emerging speech and 3D-gesture prototypes (Condon, 1988; Kendon, 1980; McNeill, 1992), which will be discussed further later in this chapter. Among the earlier systems to begin processing manual pointing or 3D gestures combined with speech were developed by Koons and colleagues (Koons, Sparrell, & Thorisson, 1993), Sharma and colleagues (Sharma et al., 1996), Poddar and colleagues (Poddar, Sethi, Ozyildiz, & Sharma, 1998), and by Duncan and colleagues (Duncan, Brown, Esposito, Holmback, & Xue, 1999).

Historically, multimodal speech and lip movement research has been driven by cognitive science interest in intersensory audio-visual perception and the coordination of speech output with lip and facial movements (Benoit & Le Goff, 1998; Bernstein & Benoit, 1996; Cohen & Massaro, 1993; Massaro & Stork, 1998; McGrath & Summerfield, 1985; McGurk & MacDonald, 1976; McLeod & Summerfield, 1987; Robert-Ribes, Schwartz, Lallouache, & Escudier, 1998; Sumby & Pollack, 1954; Summerfield, 1992; Vatikiotis-Bateson, Munhall, Hirayama, Lee, & Terzopoulos, 1996). Among the many contributions of this literature has been a detailed classification of human lip movements (visemes) and the viseme-phoneme mappings that occur during articulated speech. Existing systems that have processed combined

[1]However, other recent pen/voice multimodal systems that emphasize mobile processing, such as MiPad and the Field Medic Information System (Holzman, 1999; Huang et al., 2000), still limit pen input to pointing.

Multimodal System Characteristics:	QuickSet	Human-Centric Word Processor	VR Aircraft Maintenance Training	Field Medic Information	Portable Voice Assistant
Recognition of simultaneous or alternative individual modes	Simultaneous & individual modes	Simultaneous & individual modes	Simultaneous & individual modes	Alternative individual modes[1]	Simultaneous & individual modes
Type & size of gesture vocabulary	Pen input, Multiple gestures, Large vocabulary	Pen input, Deictic selection	3D manual input, Multiple gestures, Small vocabulary	Pen input, Deictic selection	Pen input Deictic selection[2]
Size of speech vocabulary[3] & type of linguistic processing	Moderate vocabulary, Grammar-based	Large vocabulary, Statistical language processing	Small vocabulary, Grammar-based	Moderate vocabulary, Grammar-based	Small vocabulary, Grammar-based
Type of signal fusion	Late semantic fusion, Unification, Hybrid symbolic/statistical MTC framework	Late semantic fusion, Frame-based	Late semantic fusion, Frame-based	No mode fusion	Late semantic fusion, Frame-based
Type of platform & applications	Wireless handheld, Varied map & VR applications, digital paper	Desktop computer, Word processing	Virtual reality system, Aircraft maintenance training	Wireless handheld, Medical field emergencies	Wireless handheld, Catalogue ordering
Evaluation status	Proactive user-centered design & iterative system evaluations	Proactive user-centered design	Planned for future	Proactive user-centered design & iterative system evaluations	Planned for future

[1]The FMA component recognizes speech only, and the FMC component recognizes gestural selections or speech. The FMC also can transmit digital speech and ink data, and can read data from smart cards and physiological monitors.

[2]The PVA also performs handwriting recognition.

[3]A small speech vocabulary is up to 200 words, moderate 300–1,000 words, and large in excess of 1,000 words. For pen-based gestures, deictic selection is an individual gesture, a small vocabulary is 2–20 gestures, moderate 20–100, and large in excess of 100 gestures.

FIGURE 21.1. Examples of functionality, architectural features, and general classification of different speech and gesture multimodal applications.

speech and lip movements include the classic work by Petajan (1984), Brooke and Petajan (1986), and others (Adjoudani & Benoit, 1995; Bregler & Konig, 1994; Silsbee & Su, 1996; Tomlinson, Russell, & Brooke, 1996). Additional examples of speech and lip movement systems, applications, and relevant cognitive science research have been detailed elsewhere (Benoit et al., 2000). Researchers in this area have been actively exploring adaptive techniques for improving system robustness, especially in noisy environmental contexts (Dupont & Luettin, 2000; Meier, Hürst, & Duchnowski, 1996; Potamianos, Neti, Gravier, & Garg, 2003; Rogozan & Deglise, 1998), which is an important future research direction. Although this literature has not emphasized the development of applications, nonetheless its quantitative modeling of synchronized phoneme/viseme patterns has been used to build animated characters that generate text-to-speech output and coordinated lip movements. These new animated characters are being used as an interface design vehicle for facilitating users' multimodal interaction with next-generation conversational interfaces (Cassell, Sullivan, Prevost, & Churchill, 2000; Cohen & Massaro, 1993).

While the main multimodal literatures to date have focused on either speech and pen input or speech and lip movements, recognition of other modes also is maturing and beginning to be integrated into new kinds of multimodal systems. In particular, there is growing interest in designing multimodal interfaces that incorporate vision-based technologies, such as interpretation of gaze, facial expressions, head nodding, gesturing, and large body movements (Flanagan & Huang, 2003; Morency, Sidner, Lee, & Darrell, 2005; Morimoto, Koons, Amir, Flickner, & Zhai, 1999; Pavlovic, Berry, & Huang, 1997; Turk & Robertson, 2000; Zhai et al., 1999). These technologies unobtrusively or *passively* monitor user behavior and need not require explicit user commands to a "computer." That contrasts with *active input modes*, such as speech or pen, which the user deploys intentionally as a command issued to the system (see Fig. 21.2). While passive modes may be "attentive" and less obtrusive, active modes generally are more reliable indicators of user intent.

As vision-based technologies mature, one important future direction will be the development of *blended* multimodal interfaces that combine both passive and active modes. These interfaces typically will be *temporally cascaded*, so one goal in designing new prototypes will be to determine optimal processing strategies for using advance information from the first mode (e.g., gaze) to constrain accurate interpretation of the following modes (e.g., gesture, speech). This kind of blended multimodal interface potentially can provide users with greater transparency and control, while also supporting improved robustness and broader application functionality (Oviatt & Cohen, 2000; Zhai et al., 1999). As this collection of technologies matures, there also is strong interest in designing new types of pervasive and mobile interfaces, including ones capable of adaptive processing to the user and environmental context.

As multimodal interfaces gradually evolve toward supporting more advanced recognition of users' natural activities in context, they will expand beyond rudimentary bimodal systems to ones that incorporate three or more input modes, qualitatively different modes, and more sophisticated models of multimodal interaction. This trend already has been initiated within biometrics research, which has combined recognition of multiple behavioral input modes (e.g., voice, handwriting) with physiological ones (e.g., retinal scans, fingerprints) to achieve reliable person identification and verification in challenging field conditions (Choudhury, Clarkson, Jebara, & Pentland, 1999; Jain et al., 1999; Jain & Ross, 2002; Pankanti et al., 2000).

Apart from these developments within research-level systems, multimodal interfaces also are being commercialized as products, especially in areas like personal information access and management on handhelds and cell phones. Microsoft's handheld Mipad for personal information management, and Kirusa's cell phone interface for directory assistance, messaging, and so on, are just two examples of the many mobile commercial products that are being developed with multimodal interfaces. Both include spoken language processing and a stylus for tapping on fields to constrain and guide the natural language processing. In some cases, keyboard input is supported as a third option, as well as multimedia output in the form of visualizations and text-to-speech. Another visible growth area for multimodal interfaces involves in-vehicle control of navigation, communication, and entertainment systems, which has emerged in both domestic and import cars. Mobile map-based systems and systems for safety-critical medical and military applications also are being commercialized by companies like Natural Interaction Systems (e.g., Rasa, NISMap, and NISChart applications), which places an emphasis on developing tangible multimodal interfaces that preserve users' existing work practice, minimize cognitive load, and provide backups in case of system failure (Cohen & McGee, 2004; McGee, 2003).

WHAT ARE THE GOALS AND ADVANTAGES OF MULTIMODAL INTERFACE DESIGN?

Over the past decade, numerous advantages of multimodal interface design have been documented. Unlike a traditional keyboard-and-mouse interface or a unimodal recognition-based interface, multimodal interfaces permit flexible use of input modes. This includes the choice of which modality to use for conveying different types of information, to use combined input modes, or to alternate between modes at any time. Since individual input modalities are well suited in some situations, and less ideal or even inappropriate in others, modality choice is an important design issue in a multimodal system. As systems become more complex and multifunctional, a single modality simply does not permit all users to interact effectively across all tasks and environments.

Since there are large individual differences in ability and preference to use different modes of communication, a multimodal interface permits diverse user groups to exercise selection and control over how they interact with the computer (Fell et al., 1994; Karshmer & Blattner, 1998). In this respect, multimodal interfaces have the potential to accommodate a broader range of users than traditional interfaces, including users of different ages, skill levels, native language status, cognitive styles, sensory impairments, and other temporary illnesses or permanent handicaps. For example, a visually impaired user or one with repetitive stress injury may prefer speech input and text-to-speech output. In contrast, a user with a hearing impairment

Multimodal interfaces process two or more combined user input modes— such as speech, pen, touch, manual gestures, gaze, and head and body movements— in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language and behavior, and which incorporate one or more recognition-based technologies (e.g., speech, pen, vision).

Active input modes are ones that are deployed by the user intentionally as an explicit command to a computer system (e.g., speech).

Passive input modes refer to naturally occurring user behavior or actions that are recognized by a computer (e.g., facial expressions, manual gestures). They involve user input that is unobtrusively and passively monitored, without requiring any explicit command to a computer.

Blended multimodal interfaces are ones that incorporate system recognition of at least one passive and one active input mode. (e.g., speech and lip movement systems).

Temporally-cascaded multimodal interfaces are ones that process two or more user modalities that tend to be sequenced in a particular temporal order (e.g., gaze, gesture, speech), such that partial information supplied by recognition of an earlier mode (e.g., gaze) is available to constrain interpretation of a later mode (e.g., speech). Such interfaces may combine only active input modes, only passive ones, or they may be blended.

Mutual disambiguation involves disambiguation of signal or semantic-level information in one error-prone input mode from partial information supplied by another. Mutual disambiguation can occur in a multimodal architecture with two or more semantically rich recognition-based input modes. It leads to recovery from unimodal recognition errors within a multimodal architecture, with the net effect of suppressing errors experienced by the user.

Simultaneous integrator refers to a user who habitually presents two input signals (e.g., speech, pen) in a temporally overlapped manner when communicating multimodal commands to a system

Sequential integrator refers to a user who habitually separates their multimodal signals, presenting one before the other with a brief pause intervening

Multimodal hypertiming refers to the fact that both sequential and simultaneous integrators will further accentuate their basic multimodal integration pattern when under duress (e.g., as task difficulty or system recognition errors increase)

Visemes refers to the detailed classification of visible lip movements that correspond with consonants and vowels during articulated speech. A *viseme-phoneme mapping* refers to the correspondence between visible lip movements and audible phonemes during continuous speech.

Feature-level fusion is a method for fusing low-level feature information from parallel input signals within a multimodal architecture, which has been applied to processing closely synchronized input such as speech and lip movements.

Semantic-level fusion is a method for integrating semantic information derived from parallel input modes in a multimodal architecture, which has been used for processing speech and gesture input.

FIGURE 21.2. Multimodal interface terminology.

or accented speech may prefer touch, gesture, or pen input. The natural alternation between modes that is permitted by a multimodal interface also can be effective in preventing overuse and physical damage to any single modality, especially during extended periods of computer use (Markinson[2], personal communication, 1993).

Multimodal interfaces also provide the adaptability that is needed to accommodate the continuously changing conditions of mobile use. In particular, systems involving speech, pen, or touch input are suitable for mobile tasks and, when combined, users can shift among these modalities from moment to moment as environmental conditions change (Holzman, 1999; Oviatt, 2000b, 2000c). There is a sense in which mobility can induce a state of temporary disability, such that a person is unable to use a particular input mode for some period. For example, the user of an in-vehicle application may frequently be unable to use manual or gaze input, although speech is relatively more available. In this respect, a multimodal interface permits the modality choice and switching that is needed during the changing environmental circumstances of actual field and mobile use.

A large body of data documents that multimodal interfaces satisfy higher levels of user preference when interacting with

[2]R. Markinson, University of California at San Francisco Medical School, 1993.

simulated or real computer systems. Users have a strong preference toward interacting multimodally, rather than unimodally, across a wide variety of different application domains, although this preference is most pronounced in spatial domains (Hauptmann, 1989; Oviatt, 1997). For example, 95% to 100% of users preferred to interact multimodally when they were free to use either speech or pen input in a map-based spatial domain (Oviatt, 1997). During pen/voice multimodal interaction, users preferred speech input for describing objects and events, sets and subsets of objects, out-of-view objects, conjoined information, and past and future temporal states, and for issuing commands for actions or iterative actions (Cohen & Oviatt, 1995; Oviatt & Cohen, 1991). However, their preference for pen input increased when conveying digits, symbols, graphic content, and especially when conveying the location and form of spatially oriented information on a dense graphic display such as a map (Oviatt & Olsen, 1994; Oviatt, 1997; Suhm, 1998). Likewise, 71% of users combined speech and manual gestures multimodally, rather than using one input mode, when manipulating graphic objects on a CRT screen (Hauptmann, 1989).

During the early design of multimodal systems, it was assumed that efficiency gains would be the main advantage of designing an interface multimodally, and that this advantage would derive from the ability to process input modes in parallel. It is true that multimodal interfaces sometimes support improved efficiency, especially when manipulating graphical information. In simulation research comparing speech-only with multimodal pen/voice interaction, empirical work demonstrated that multimodal interaction yielded 10% faster task-completion time during visual-spatial tasks, but no significant efficiency advantage in verbal or quantitative task domains (Oviatt, 1997; Oviatt, Cohen, & Wang, 1994). Likewise, users' efficiency improved when they combined speech and gestures multimodally to manipulate 3D objects, compared with unimodal input (Hauptmann, 1989). In another early study, multimodal speech-and-mouse input improved efficiency in a line-art drawing task (Leatherby & Pausch, 1992). Finally, in a study that compared task-completion times for a graphical interface versus a multimodal pen/voice interface, military domain experts averaged four times faster at setting up complex simulation scenarios on a map when they were able to interact multimodally (Cohen, McGee, & Clow, 2000). This latter study was based on testing of a fully functional multimodal system, and it included time required to correct recognition errors.

One particularly advantageous feature of multimodal interface design is its superior error handling, both in terms of error avoidance and graceful recovery from errors (Oviatt & van Gent, 1996; Oviatt, Bernard, & Levow, 1999; Oviatt, 1999a; Rudnicky & Hauptmann, 1992; Suhm, 1998; Tomlinson et al., 1996). There are user-centered and system-centered reasons why multimodal systems facilitate error recovery, when compared with unimodal recognition-based interfaces. For example, in a multimodal speech and pen-based gesture interface users will select the input mode that they judge to be less error prone for particular lexical content, which tends to lead to error avoidance (Oviatt & van Gent, 1996). They may prefer speedy speech input, but will switch to pen input to communicate a foreign surname. Secondly, users' language often is simplified when interacting multimodally, which can substantially reduce the complexity of natural language processing and thereby reduce recognition errors, as described later in this chapter (Oviatt & Kuhn, 1998). In one study, users' multimodal utterances were documented to be briefer, to contain fewer complex locative descriptions, and 50% fewer spoken disfluencies, when compared with a speech-only interface. Thirdly, users have a strong tendency to switch modes after system recognition errors, which facilitates error recovery. This error resolution occurs because the confusion matrices differ for any given lexical content for the different recognition technologies involved in processing (Oviatt et al., 1999; Oviatt, 2002).

In addition to these user-centered reasons for better error avoidance and resolution, there also are system-centered reasons for superior error handling. A well-designed multimodal architecture with two semantically rich input modes can support *mutual disambiguation* of input signals. For example, if a user says "ditches" but the speech recognizer confirms the singular "ditch" as its best guess, then parallel recognition of several graphic marks can result in recovery of the correct plural interpretation. This recovery can occur in a multimodal architecture even though the speech recognizer initially ranks the plural interpretation "ditches" as a less preferred choice on its n-best list. Mutual disambiguation involves recovery from unimodal recognition errors within a multimodal architecture, because semantic information from each input mode supplies partial disambiguation of the other mode, thereby leading to more stable and robust overall system performance (Oviatt, 1999a, 2000a, 2002). Another example of mutual disambiguation is shown in Fig. 21.3. To achieve optimal error handling, a multimodal interface ideally should be designed to include complementary input modes, and

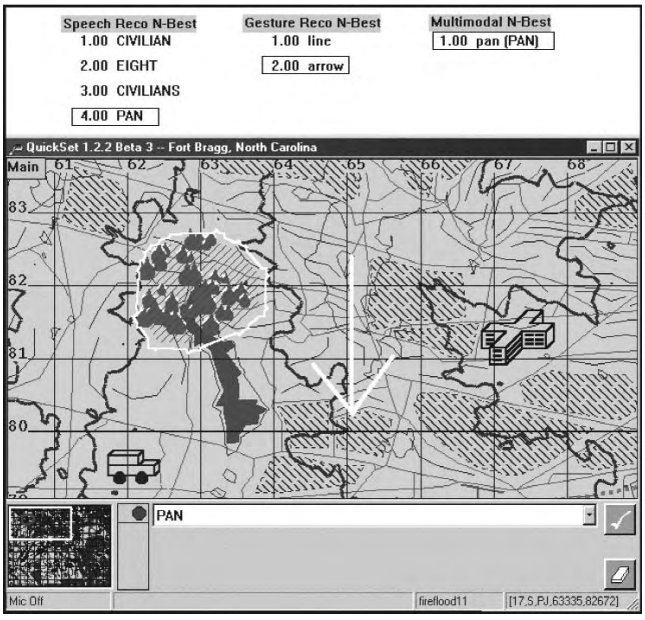


FIGURE 21.3. Multimodal command to "pan" the map, which illustrates mutual disambiguation occurring between incoming speech and gesture information, such that lexical hypotheses were pulled up on both *n*-best lists to produce a correct final multimodal interpretation.

the alternative input modes should provide duplicate functionality such that users can accomplish their goals using either mode.

In two recent studies involving over 4,600 multimodal commands, a multimodal architecture was found to support mutual disambiguation and error suppression ranging between 19% and 41% (Oviatt, 1999a, 2000a, 2002). Improved robustness also was greater for "challenging" user groups (accented vs. native speakers) and usage contexts (mobile vs. stationary use). These results indicate that a well-designed multimodal system not only can perform more robustly than a unimodal system, but also in a more stable way across varied real-world users and usage contexts. Finally, during audio-visual perception of speech and lip movements, improved speech recognition also has been demonstrated for both human listeners (McLeod & Summerfield, 1987) and multimodal systems (Adjoudani & Benoit, 1995; Tomlinson et al., 1996).

Another recent focus has been on the advantages of multimodal interface design for minimizing users' cognitive load. As task complexity increases, there is evidence that users self-manage their working memory limits by distributing information across multiple modalities, which in turn enhances their task performance during both perception and production (Calvert, Spence, & Stein, 2004; Mousavi, Low, & Sweller, 1995; Oviatt, 1997; Oviatt, Coulston, & Lunsford, 2004; Tang, McLachlan, Lowe, Saka, & MacLean, 2005). These predictions and findings are based on Wickens and colleagues' cognitive resource theory and Baddeley's theory of working memory (Baddeley, 1992; Wickens, Sandry, & Vidulich, 1983). The latter maintains that short-term or working memory consists of multiple independent processors associated with different modes. This includes a visual-spatial "sketch pad" that maintains visual materials such as pictures and diagrams in one area of working memory, and a separate phonological loop that stores auditory-verbal information. Although these two processors are believed to be coordinated by a central executive, in terms of lower-level modality processing they are viewed as functioning largely independently, which is what enables the effective size of working memory to expand when people use multiple modalities during tasks (Baddeley, 1992). So with respect to management of cognitive load, the inherent flexibility of multimodal interfaces is well suited to accommodating the high and changing load con ditions typical of realistic mobile use.

WHAT METHODS AND INFORMATION HAVE BEEN USED TO DESIGN NOVEL MULTIMODAL INTERFACES?

The design of new multimodal systems has been inspired and organized largely by two things. First, the cognitive science literature on intersensory perception and intermodal coordination during production is beginning to provide a foundation of information for user modeling, as well as information on what systems must recognize and how multimodal architectures should be organized. For example, the cognitive science literature has provided knowledge of the natural integration patterns that typify people's lip and facial movements with speech output (Benoit, Guiard-Marigny, Le Goff, & Adjoudani, 1996; Ekman, 1992; Ekman & Friesen, 1978; Fridlund, 1994; Hadar, Steiner, Grant, & Rose, 1983; Massaro & Cohen, 1990; Stork & Hennecke, 1995; Vatikiotis-Bateson et al., 1996), and their coordinated use of manual or pen-based gestures with speech (Kendon, 1980; McNeill, 1992; Oviatt, DeAngeli, & Kuhn, 1997). Given the complex nature of users' multimodal interaction, cognitive science has played and will continue to play an essential role in guiding the design of robust multimodal systems. In this respect, a multidisciplinary perspective will be more central to successful multimodal system design than it has been for traditional GUI design. The cognitive science underpinnings of multimodal system design are described later in this chapter.

Secondly, high-fidelity automatic simulations also have played a critical role in prototyping new types of multimodal systems (Dahlbäck, Jëonsson, & Ahrenberg, 1992; Oviatt et al., 1992). When a new multimodal system is in the planning stages, design sketches and low-fidelity mock-ups may initially be used to visualize the new system and plan the sequential flow of human-computer interaction. These tentative design plans then are rapidly transitioned into a higher-fidelity simulation of the multimodal system, which is available for proactive and situated data collection with the intended user population. High-fidelity simulations have been the preferred method for designing and evaluating new multimodal systems, and extensive data collection with such tools preferably is completed before a fully functional system ever is built.

During high-fidelity simulation testing, a user interacts with what she believes is a fully functional multimodal system although the interface is actually a simulated front-end designed to appear and respond as the fully functional system would. During the interaction, a programmer assistant at a remote location provides the simulated system responses. As the user interacts with the front end, the programmer tracks her multimodal input and provides system responses as quickly and accurately as possible. To support this role, the programmer makes use of automated simulation software that is designed to support interactive speed, realism with respect to the targeted system, and other important characteristics. For example, with these automated tools, the programmer may be able to make a single selection on a workstation field to rapidly send simulated system responses to the user during a data-collection session.

High-fidelity simulations have been the preferred method for prototyping multimodal systems for several reasons. Simulations are relatively easy and inexpensive to adapt, compared with building and iterating a complete system. They also permit researchers to alter a planned system's characteristics in major ways (e.g., input and output modes available), and to study the impact of different interface features in a systematic and scientific manner (e.g., type and base-rate of system errors). In comparison, a particular system with its fixed characteristics is a less flexible and suitable research tool, and the assessment of any single system basically amounts to an individual case study. Using simulation techniques, rapid adaptation and investigation of planned system features permit researchers to gain a broader and more principled perspective on the potential of newly emerging technologies. In a practical sense, simulation research can assist in the evaluation of critical performance tradeoffs and in making decisions about alternative system designs, which designers must do as they strive to create more usable multimodal systems.

To support the further development and commercialization of multimodal systems, additional infrastructure that will be needed in the future includes (a) simulation tools for rapidly building and reconfiguring multimodal interfaces, (b) automated tools for collecting and analyzing multimodal corpora, and (c) automated tools for iterating new multimodal systems to improve their performance (see Oviatt et al., 2000, for further discussion).

WHAT ARE THE COGNITIVE SCIENCE UNDERPINNINGS OF MULTIMODAL INTERFACE DESIGN?

This section discusses the growing cognitive science literature that provides the empirical underpinnings needed to design next-generation multimodal interfaces. The ability to develop multimodal systems depends on knowledge of the natural integration patterns that typify people's combined use of different input modes. In particular, the design of new multimodal systems depends on intimate knowledge of the properties of different modes and the information content they carry, the unique characteristics of multimodal language and its processability, and the integration and synchronization characteristics of users' multimodal interaction. It also relies on accurate prediction of when users are likely to interact multimodally, and how alike different users are in their specific integration patterns. The relevant cognitive science literature on these topics is very extensive, especially when consideration is given to all of the underlying sensory perception and production capabilities involved in different input modes currently being incorporated in new multimodal interfaces. As a result, this section will be limited to introducing the main cognitive science themes and findings that are relevant to the more common types of multimodal system.

This cognitive science foundation also has played a key role in identifying computational "myths" about multimodal interaction, and replacing these misconceptions with contrary empirical evidence. Figure 21.4 summarizes 10 common myths about multimodal interaction, which are addressed and discussed in more detail elsewhere (Oviatt, 1999b). As such, the literature summarized in this section aims to provide a more accurate foundation for guiding the design of next-generation multimodal systems.

Ten Myths of Multimodal Interaction

Myth #1: *If you build a multimodal system, users will interact multimodally*

Myth #2: *Speech & pointing is the dominant multimodal integration pattern*

Myth #3: *Multimodal input involves simultaneous signals*

Myth #4: *Speech is the primary input mode in any multimodal system that includes it*

Myth #5: *Multimodal language does not differ linguistically from unimodal language*

Myth #6: *Multimodal integration involves redundancy of content between modes*

Myth #7: *Individual error-prone recognition technologies combine multimodally to produce even greater unreliability*

Myth #8: *All users' multimodal commands are integrated in a uniform way*

Myth #9: *Different input modes are capable of transmitting comparable content*

Myth #10: *Enhanced efficiency is the main advantage of multimodal systems*

(taken from Oviatt, 1999b)

FIGURE 21.4. Ten myths of multimodal interaction: Separating myth from empirical reality.

When Do Users Interact Multimodally?

During natural interpersonal communication, people are always interacting multimodally. Of course, in this case the number of information sources or modalities that an interlocutor has available to monitor is essentially unlimited. However, all multimodal systems are constrained in the number and type of input modes they can recognize. Also, a user can compose active input during human-computer interaction that either is delivered multimodally or that is delivered entirely using just one mode. That is, although users in general may have a strong preference to interact multimodally rather than unimodally, this is no guarantee that they will issue every command to a system multimodally, given the particular type of multimodal interface available. Therefore, the first nontrivial question that arises during system processing is whether a user is communicating unimodally or multimodally.

In the case of speech- and pen-based multimodal systems, users typically mix unimodal and multimodal expressions. In one study involving a visual-spatial domain, users' commands were expressed multimodally 20% of the time, with others just spoken or written (Oviatt et al., 1997). In contrast, in other spatial domains, the ratio of users' multimodal interaction often is 65% to 70% (Oviatt, 1999b; Oviatt et al., 2004). Predicting whether a user will express a command multimodally also depends on the type of action she is performing. In particular, users usually express commands multimodally when describing spatial information about the location, number, size, orientation, or shape of an object. In one study, users issued multimodal commands 86% of the time when they had to add, move, modify, or calculate the distance between objects on a map in a way that required specifying spatial locations (Oviatt et al., 1997). They also were moderately likely to interact multimodally when selecting an object from a larger array, for example, when deleting a particular object from the map. However, when performing general actions without any spatial component, such as printing a map, users expressed themselves multimodally less than 1% of the time. These data emphasize that future multimodal systems will need to distinguish between instances in which users are and are not communicating multimodally, so that accurate decisions can be made about when parallel input streams should be interpreted jointly versus individually. They also suggest that knowledge of the type of actions to be included in an application, such as whether the application entails manipulating spatial information, should influence the basic decision of whether to build a multimodal interface at all.

Findings from a more recent study reveal that multimodal interface users spontaneously respond to dynamic changes in their own cognitive load by shifting to multimodal communication as load increases with task difficulty and communicative complexity (Oviatt, Coulston, & Lunsford, 2004). Given a flexible multimodal interface, users' ratio of multimodal (versus

unimodal) interaction increased substantially from 18.6% when referring to established dialogue context to 77.1% when required to establish a new context, a +315% relative increase. Likewise, the ratio of users' multimodal interaction increased significantly as the tasks became more difficult, from 59.2% during low difficulty tasks, to 65.5% at moderate difficulty, 68.2% at high, and 75.0% at very high difficulty, an overall relative increase of +27%. These adaptations in multimodal interaction levels reflect users' efforts to self-manage limitations in their working memory as discourse-level demands and task complexity increased. As discussed earlier, users accomplished this by distributing communicative information across multiple modalities in a manner compatible with a cognitive load theory of multimodal interaction. This interpretation is consistent with Baddeley's theory of working memory (Baddeley, 1992), as well as the growing literatures within education (Mousavi, Low, & Sweller, 1995; Sweller, 1988), linguistics (Almor, 1999), and multisensory perception (Calvert, Spence, & Stein, 2004; Ernst & Bulthoff, 2004). Recent work on visual and haptic processing under workload also indicates that presentation of haptic feedback during a complex task can augment users' ability to handle visual information overload (Tang, McLachlan, Lowe, Saka, & MacLean, 2005).

In a multimodal interface that processes passive or blended input modes, there always is at least one passively tracked input source providing continuous information (e.g., gaze tracking, head position). In these cases, all user input would by definition be classified as multimodal, and the primary problem would become segmentation and interpretation of each continuous input stream into meaningful actions of significance to the application. In the case of blended multimodal interfaces (e.g., gaze tracking and mouse input), it still may be opportune to distinguish active forms of user input that might be more accurately or expeditiously handled as unimodal events.

What Are the Integration and Synchronization Characteristics of Users' Multimodal Input?

The past literature on multimodal systems has focused largely on simple selection of objects or locations in a display, rather than considering the broader range of multimodal integration patterns. Since the development of Bolt's (1980) "Put That There" system, speak-and-point has been viewed as the prototypical form of multimodal integration. In Bolt's system, semantic processing was based on spoken input, but the meaning of a deictic term such as "that" was resolved by processing the *x/y* coordinate indicated by pointing at an object. Since that time, other multimodal systems also have attempted to resolve deictic expressions using a similar approach—for example, using gaze location instead of manual pointing (Koons et al., 1993).

Unfortunately, this concept of multimodal interaction as point-and-speak makes only limited use of new input modes for selection of objects—just as the mouse does. In this respect, it represents the persistence of an old mouse-oriented metaphor. In contrast, modes that transmit written input, manual gesturing, and facial expressions are capable of generating symbolic information that is much more richly expressive than simple pointing or selection. In fact, studies of users' integrated pen/voice input indicate that a speak-and-point pattern only comprises 14% of all spontaneous multimodal utterances (Oviatt et al., 1997). Instead, pen input more often is used to create graphics, symbols and signs, gestural marks, digits, and lexical content. During interpersonal multimodal communication, linguistic analysis of spontaneous manual gesturing also indicates that simple pointing accounts for less than 20% of all gestures (McNeill, 1992). Together, these cognitive science and user-modeling data highlight the fact that any multimodal system designed exclusively to process speak-and-point will fail to provide users with much useful functionality. For this reason, specialized algorithms for processing deictic-point relations will have only limited practical use in the design of future multimodal systems. It is clear that a broader set of multimodal integration issues needs to be addressed in future work. Future research also should explore typical integration patterns between other promising modality combinations, such as speech and gaze.

It also is commonly assumed that any signals involved in a multimodal construction will co-occur temporally. The presumption is that this temporal overlap then determines which signals to combine during system processing. In the case of speech and manual gestures, successful processing of the deictic term "that square" in Bolt's original system relied on interpretation of pointing when the word "that" was spoken in order to extract the intended referent. However, one empirical study indicated that users often do not speak deictic terms at all, and when they do, the deictic frequently is not overlapped in time with their pointing. In fact, it has been estimated that as few as 25% of users' commands actually contain a spoken deictic that overlaps with the pointing needed to disambiguate its meaning (Oviatt et al., 1997).

Beyond the issue of deixis, a series of studies has shown that users' input frequently does not overlap at all during multimodal commands to a computer (Oviatt, 1999b; Oviatt et al., 2003; Oviatt et al., 2005; Xiao, Girand, & Oviatt, 2002; Xiao, Lunsford, Coulston, Wesson, & Oviatt, 2003). In fact, there are two distinct types of user with respect to integration patterns: *simultaneous* integrators and *sequential* ones. A user who habitually integrates her speech and pen input in a *simultaneous* manner overlaps them temporally, whereas a sequential integrator finishes one mode before beginning the second, as summarized in Fig. 21.2. These two types of user integration patterns occur across the lifespan from children through the elderly (Oviatt et al., 2005; Xiao et al., 2002; Xiao et al., 2003). They also can be detected almost immediately during multimodal interaction, usually on the very first input. Users' habitual integration pattern remains strikingly highly consistent during a session, as well as resistant to change following explicit instructions or attempts at training (Oviatt et al., 2003; Oviatt et al., 2005). This bimodal distribution of user integration patterns has been observed in different task domains (e.g., map-based real estate selection, crisis management, educational applications with animated characters), and also when using different types of interface (e.g., conversational, command style) (Oviatt, 1999b; Xiao et al., 2002; Xiao et al., 2003). In short, empirical studies have demonstrated that this bimodal distinction between users in their fundamental integration pattern generalizes widely across different age groups, task domains, and types of interface.

One interesting discovery in recent work is the phenomenon of *multimodal hypertiming*, which refers to the fact that both sequential and simultaneous integrators will entrench further or accentuate their habitual multimodal integration pattern (*e.g.*, increasing their intermodal *lag* during sequential integrations, or *overlap* during simultaneous integrations, as summarized in 21.2) during system error handling or when completing increasingly difficult tasks. In fact, users will progressively increase their degree of entrenchment by 18% as system errors increase, and by 59% as task difficulty increases (Oviatt et al., 2003). As such, changes in the degree of users' multimodal hypertiming provide a potentially sensitive means of evaluating their cognitive load during real-time interactive exchanges. In the context of system error handling, the phenomenon of multimodal hypertiming basically replaces the hyperarticulation that is typically observed in users during error-prone speech-only interactions.

Given the bimodal distribution of user integration patterns, *adaptive temporal thresholds* potentially could support more tailored and flexible approaches to fusion. Ideally, an adaptive multimodal system would detect, automatically learn, and adapt to a user's dominant multimodal integration pattern, which could result in substantial improvements in system processing speed, accuracy of interpretation, and synchronous interchange with the user. For example, it has been estimated that system delays could be reduced to approximately 40% to 50% of what they currently are by adopting user-defined thresholds (Oviatt et al., 2005). Recent research has begun comparing different learning-based models for adapting a multimodal system's temporal thresholds to an individual user in real time (Huang & Oviatt, in press).

Unfortunately, users' multimodal integration patterns have not been studied as extensively or systematically for other input modes, such as speech and manual gesturing. Linguistics research on interpersonal communication patterns has revealed that both spontaneous gesturing and signed language often precede their spoken lexical analogues during human communication (Kendon, 1980; Naughton, 1996), when considering word-level integration pattern. In fact, the degree to which gesturing precedes speech is greater in topic-prominent languages such as Chinese than it is in subject-prominent ones like Spanish or English (McNeill, 1992). Even in the speech and lip-movement literature, close but not perfect temporal synchrony is typical, with lip movements occurring a fraction of a second before the corresponding auditory signal (Abry, Lallouache, & Cathiard, 1996; Benoit, 2000). However, when considering the whole user utterance as the unit of analysis, some other studies of speech and manual gesturing have found a higher rate of simultaneity for these modes (Epps, Oviatt, & Chen, 2004). Learning-based approaches that are capable of accurately identifying and adapting to different multimodal integration patterns, whether due to differences among users, modality combinations, or applications and usage contexts, will be required in order to generalize and speed up multimodal system development in the future.

In short, although two input modes may be highly interdependent and synchronized during multimodal interaction, synchrony does not imply simultaneity. The empirical evidence reveals that multimodal signals often do not co-occur temporally at all during human-computer or natural human communication. Therefore, multimodal system designers cannot necessarily count on conveniently overlapped signals in order to achieve successful processing in the multimodal architectures they build. Future research needs to explore the integration patterns and temporal cascading that can occur among three or more input modes—such as gaze, gesture, and speech—so that more advanced multimodal systems can be designed and prototyped.

In the design of new multimodal architectures, it is important to note that data on the order of input modes and average time lags between input modes has been used to determine the likelihood that an utterance is multimodal versus unimodal, and to establish temporal thresholds for fusion of input. In the future, weighted likelihoods associated with different utterance segmentations, for example, that an input stream containing speech, writing, speech should be segmented into [S/W S] rather than [S W/S], and with intermodal time lag distributions, will be used to optimize correct recognition of multimodal user input (Oviatt, 1999b). In the design of future time-critical multimodal architectures, data on users' integration and synchronization patterns will need to be collected for other mode combinations during realistic interactive tasks, so that temporal thresholds can be established for performing multimodal fusion.

What Individual Differences Exist in Multimodal Interaction, and What Are the Implications for Designing Systems for Universal Access?

There are large individual differences in users' multimodal interaction patterns, beginning with their overall preference to interact unimodally versus multimodally, and which mode they generally prefer (e.g., speaking versus writing) (Oviatt et al., 2004). As outlined above, there likewise are striking differences among users in adopting either a sequential or simultaneous multimodal integration pattern. Recent research has revealed that these two patterns are associated with behavioral and linguistic differences between the groups (Oviatt et al., 2005). Whereas in an interactive task context their performance speed was comparable, sequential integrators were far less error-prone and excelled during new or complex tasks. Although their speech rate was no slower, sequential integrators also had more precise articulation (e.g., less disfluency). Finally, sequential integrators were more likely to adopt terse and direct command-style language, with a smaller and less varied vocabulary, which appeared focused on achieving error-free communication. These user differences in interaction patterns have been interpreted as deriving from fundamental differences among users in their reflective-impulsive cognitive style (Oviatt et al., 2005). Based on this work, one goal of future multimodal interface design will be to support the poorer attention span and higher error rate of impulsive users—especially for mobile in-vehicle, military, and similar application contexts in which the cost of committing errors is unacceptably high.

Apart from these individual differences, cultural differences also have been documented between users in modality integration patterns. For example, substantial individual differences have been reported in the temporal synchrony between speech and lip movements (Kricos, 1996) and, in addition, lip movements during speech production are known to be less exaggerated

among Japanese speakers than Americans (Sekiyama & Tohkura, 1991). In fact, extensive inter-language differences have been observed in the information available from lip movements during audio-visual speech (Fuster-Duran, 1996). These findings have implications for the degree to which disambiguation of speech can be achieved through lip movement information in noisy environments or for different user populations. Finally, nonnative speakers, the hearing impaired, and elderly listeners all are more influenced by visual lip movement than auditory cues when processing speech (Fuster-Duran, 1996; Massaro, 1996). These results have implications for the design and expected value of audio-visual multimedia output for different user groups in animated character interfaces. With respect to support for universal access, recent work also has shown the advantage of combined audio-visual processing for recognition of impaired speech (Potamianos & Neti, 2001).

Finally, gender, age, and other individual differences are common in gaze patterns, as well as speech and gaze integration (Argyle, 1972). As multimodal interfaces incorporating gaze become more mature, further research will need to explore these gender and age-specific patterns, and to build appropriately adapted processing strategies. In summary, considerably more research is needed on multimodal integration and synchronization patterns for new mode combinations, as well as for diverse and disabled users for whom multimodal interfaces may be especially suitable for ensuring universal access.

Is Complementarity or Redundancy the Main Organizational Theme That Guides Multimodal Integration?

It frequently is claimed that the propositional content conveyed by different modes during multimodal communication contains a high degree of redundancy. However, the dominant theme in users' natural organization of multimodal input actually is complementarity of content, not redundancy. For example, speech and pen input consistently contribute different and complementary semantic information, with the subject, verb, and object of a sentence typically spoken, and locative information written (Oviatt et al., 1997). In fact, a major complementarity between speech and manually oriented pen input involves visual-spatial semantic content, which is one reason these modes are an opportune combination for visual-spatial applications. Whereas spatial information is uniquely and clearly indicated via pen input, the strong descriptive capabilities of speech are better suited for specifying temporal and other nonspatial information. Even during multimodal correction of system errors, when users are highly motivated to clarify and reinforce their information delivery, speech and pen input express redundant information less than 1% of the time. Finally, during interpersonal communication linguists also have documented that spontaneous speech and manual gesturing involve complementary rather than duplicate information between modes (McNeill, 1992).

Other examples of primary multimodal complementarities during interpersonal and human-computer communication have been described in past research (McGurk & MacDonald, 1976; Oviatt & Olsen, 1994; Wickens et al., 1983). For example, in the literature on multimodal speech and lip movements, natural feature-level complementarities have been identified between visemes and phonemes for vowel articulation, with vowel rounding better conveyed visually, and vowel height and backness better revealed auditorally (Massaro & Stork, 1998; Robert-Ribes et al., 1998).

In short, actual data highlight the importance of complementarity as a major organizational theme during multimodal communication. The designers of next-generation multimodal systems therefore should not expect to rely on duplicated information when processing multimodal language, although in certain contexts (such as teaching) a greater percentage of duplicate content than usual may be expected to exist. In multimodal systems involving both speech and pen-based gestures and speech and lip movements, one explicit goal has been to integrate complementary modalities in a manner that yields a synergistic blend, such that each mode can be capitalized upon and used to overcome weaknesses in the other mode (Cohen et al., 1989). This approach to system design has promoted the philosophy of using modes and component technologies to their natural advantage, and of combining them in a manner that permits mutual disambiguation. One advantage of achieving such a blend is that the resulting multimodal architecture can function more robustly than an individual recognition-based technology or a multimodal system based on input modes lacking natural complementarities.

What Are the Primary Features of Multimodal Language?

Communication channels can be tremendously influential in shaping the language transmitted within them. From past research, there now is cumulative evidence that many linguistic features of multimodal language are qualitatively very different from that of spoken or formal textual language. In fact, it can differ in features as basic as brevity, semantic content, syntactic complexity, word order, disfluency rate, degree of ambiguity, referring expressions, specification of determiners, anaphora, deixis, and linguistic indirectness. In many respects, multimodal language is simpler linguistically than spoken language. In particular, comparisons have revealed that the same user completing the same map-based task communicates significantly fewer words, briefer sentences, and fewer complex spatial descriptions and disfluencies when interacting multimodally, compared with using speech alone (Oviatt, 1997). One implication of these findings is that multimodal interface design has the potential to support more robust future systems than a unimodal design approach. The following is an example of a typical user's spoken input while attempting to designate an open space using a map system: "Add an open space on the north lake to b—include the north lake part of the road and north." In contrast, the same user accomplished the same task multimodally by encircling a specific area and saying, "Open space."

In previous research, hard-to-process, disfluent language has been observed to decrease by 50% during multimodal interaction with a map, compared with a more restricted speech-only inter-

action (Oviatt, 1997). This drop occurs mainly because people have difficulty speaking spatial information, which precipitates disfluencies. In a flexible multimodal interface, they instead use pen input to convey spatial information, thereby avoiding the need to speak it. Further research is needed to establish whether other forms of flexible multimodal communication also generally ease users' cognitive load, which may be reflected in a reduced rate of disfluencies.

During multimodal pen/voice communication, the linguistic indirection that is typical of spoken language frequently is replaced with more direct commands (Oviatt & Kuhn, 1998). In the following example, a study participant made a disfluent indirect request using speech input while requesting a map-based distance calculation: "What is the distance between the Victorian Museum and the, uh, the house on the east side of Woodpecker Lane?" When requesting distance information multimodally, the same user encircled the house and museum while speaking the following brief direct command: "Show distance between here and here." In this research, the briefer and more direct multimodal pen/voice language also contained substantially fewer referring expressions, with a selective reduction in co-referring expressions that instead were transformed into deictic expressions. This latter reduction in coreference would simplify natural language processing by easing the need for anaphoric tracking and resolution in a multimodal interface. Also consistent with fewer referring expressions, explicit specification of definite and indefinite reference is less common in multimodal language (Oviatt & Kuhn, 1998). Current natural language processing algorithms typically rely heavily on the specification of determiners in definite and indefinite references in order to represent and resolve noun-phrase reference. One unfortunate byproduct of the lack of such specifications is that current language processing algorithms are unprepared for the frequent occurrence of elision and deixis in multimodal human-computer interaction.

In other respects, multimodal language clearly is different from spoken language, although not necessarily simpler. For example, users' multimodal pen/voice language departs from the canonical English word order of S-V-O-LOC (e.g., Subject-Verb-Object-Locative constituent), which is observed in spoken language and formal textual language. Instead, users' multimodal constituents shift to a LOC-S-V-O word order. A recent study reported that 95% of locative constituents were in sentence-initial position during multimodal interaction. However, for the same users completing the same tasks while speaking, 96% of locatives were in sentence-final position (Oviatt et al., 1997). It is likely that broader analysis of multimodal communication patterns, which could involve gaze and manual gesturing to indicate location rather than pen-based pointing, would reveal a similar reversal in word order.

One implication of these many differences is that new multimodal corpora, statistical language models, and natural language-processing algorithms will need to be established before multimodal language can be processed optimally. Future research and corpus-collection efforts also will be needed on different types of multimodal communication, and in other application domains, so that the generality of previously identified multimodal language differences can be explored.

WHAT ARE THE BASIC WAYS IN WHICH MULTIMODAL INTERFACES DIFFER FROM GRAPHICAL USER INTERFACES?

Multimodal research groups currently are rethinking and redesigning basic user interface architectures because a whole new range of architectural requirements has been posed. First, graphical user interfaces typically assume that a single event stream controls the underlying event loop, with any processing sequential in nature. For example, most GUIs ignore typed input when a mouse button is depressed. In contrast, multimodal interfaces typically can process continuous and simultaneous input from parallel incoming streams. Secondly, GUIs assume that the basic interface actions, such as selection of an item, are atomic and unambiguous events. In contrast, multimodal systems process input modes using recognition-based technologies, which are designed to handle uncertainty and entail probabilistic methods of processing. Thirdly, GUIs often are built to be separable from the application software that they control, although the interface components usually reside centrally on one machine. In contrast, recognition-based user interfaces typically have larger computational and memory requirements, which often makes it desirable to distribute the interface over a network so that separate machines can handle different recognizers or databases. For example, cell phones and networked PDAs may extract features from speech input, but transmit them to a recognizer that resides on a server. Finally, multimodal interfaces that process two or more recognition-based input streams require time stamping of input, and the development of temporal constraints on mode fusion operations. In this regard, they involve uniquely time-sensitive architectures.

WHAT BASIC ARCHITECTURES AND PROCESSING TECHNIQUES HAVE BEEN USED TO DESIGN MULTIMODAL SYSTEMS?

Many early multimodal interfaces that handled combined speech and gesture, such as Bolt's "Put That There" system (Bolt, 1980), have been based on a control structure in which multimodal integration occurs during the process of parsing spoken language. As discussed earlier, when the user speaks a deictic expression such as "here" or "this," the system searches for a synchronized gestural act that designates the spoken referent. While such an approach is viable for processing a point-and-speak multimodal integration pattern, as discussed earlier, multimodal systems must be able to process richer input than just pointing, including gestures, symbols, graphic marks, lip movements, meaningful facial expressions, and so forth. To support more broadly functional multimodal systems, general processing architectures have been developed since Bolt's time. Some of these recent architectures handle a variety of multimodal integration patterns, as well as the interpretation of both unimodal and combined multimodal input. This kind of architecture can support the development of multimodal systems in which modalities are processed individually as input alternatives

to one another, or those in which two or more modes are processed as combined multimodal input.

For multimodal systems designed to handle joint processing of input signals, there are two main subtypes of multimodal architecture. One subtype integrates signals at the *feature level* (e.g., "early fusion"). The other integrates information at a *semantic level* (e.g., "late fusion"). Examples of systems based on an early feature-fusion processing approach include those developed by Bregler and colleagues (Bregler, Manke, Hild, & Waibel, 1993), Vo and colleagues (Vo et al., 1995), and Pavlovic and colleagues (Pavlovic, Sharma, & Huang, 1997; Pavlovic & Huang, 1998). In feature-fusion architecture, the signal-level recognition process in one mode influences the course of recognition in the other. Feature fusion is considered more appropriate for closely temporally synchronized input modalities, such as speech and lip movements (Stork & Hennecke, 1995; Rubin et al., 1998).

In contrast, multimodal systems using the late semantic fusion approach have been applied to processing multimodal speech and pen input or manual gesturing, for which the input modes are less coupled temporally. These input modes provide different but complementary information that typically is integrated at the utterance level. Late semantic integration systems use individual recognizers that can be trained using unimodal data, which are easier to collect and are already publicly available for speech and handwriting. In this respect, systems based on semantic fusion can be scaled up more easily in number of input modes or vocabulary size. Examples of systems based on semantic fusion include Put That There (Bolt, 1980), ShopTalk (Cohen et al., 1989), QuickSet (Cohen et al., 1997), CUBRICON (Neal & Shapiro, 1991), Virtual World (Codella et al., 1992), Finger-Pointer (Fukumoto, Suenaga, & Mase, 1994), VisualMan (Wang, 1995), Human-Centric Word Processor, Portable Voice Assistant (Bers et al., 1998), the VR Aircraft Maintenance Training System (Duncan et al., 1999) and Jeanie (Vo & Wood, 1996).

As an example of multimodal information processing flow in a late-stage semantic architecture, Fig. 21.5 illustrates two input modes (e.g., speech and manual or pen-based gestures) recognized in parallel and processed by an understanding component. The results involve partial meaning representations that are fused by the multimodal integration component, which also is influenced by the system's dialogue management and interpretation of current context. During the integration process, alternative lexical candidates for the final multimodal interpretation

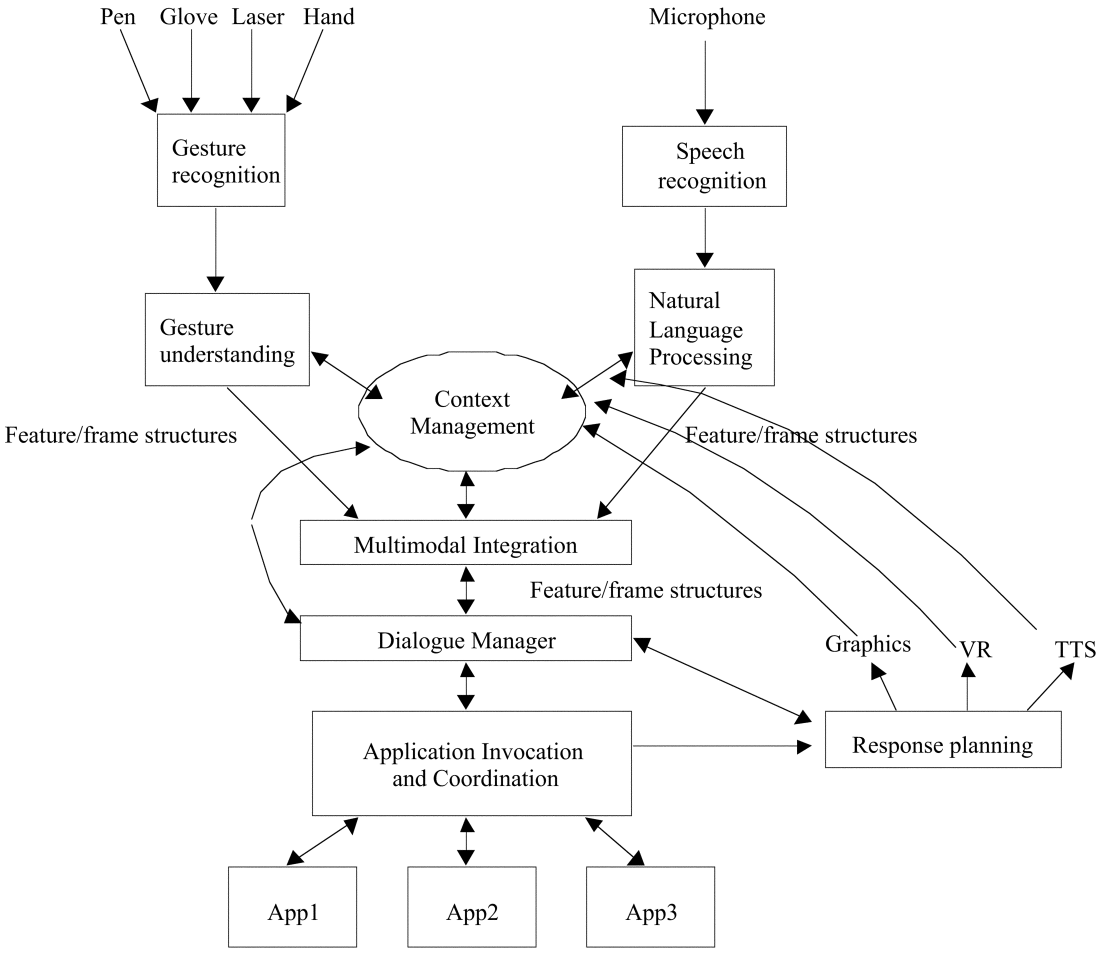


FIGURE 21.5. Typical information processing flow in a multimodal architecture designed for speech and gesture.

are ranked according to their probability estimates on an *n*-best list. The best-ranked multimodal interpretation then is sent to the application invocation and control component, which transforms this information into a series of commands to one or more back-end application systems. System feedback typically includes multimedia output, which may incorporate text-to-speech and non-speech audio; graphics and animation; and so forth. For examples of feature-based multimodal processing flow and architectures, especially as applied to multimodal speech and lip movement systems, see Benoit et al. (2000).

There are many ways to realize this information processing flow as architecture. One common infrastructure that has been adopted by the multimodal research community involves *multi-agent architectures*, such as the Open Agent Architecture (Cohen, Cheyer, Wang, & Baeg, 1994; Martin, Cheyer, & Moran, 1999) and Adaptive Agent Architecture (Kumar & Cohen, 2000). In a multi-agent architecture, the many components needed to support the multimodal system (e.g., speech recognition, gesture recognition, natural language processing, multimodal integration) may be written in different programming languages, on different machines, and with different operating systems. Agent communication languages are being developed that can handle asynchronous delivery, triggered responses, multi-casting, and other concepts from distributed systems, and that are fault-tolerant (Kumar & Cohen, 2000). Using a multi-agent architecture, for example, speech and gestures can arrive in parallel or asynchronously via individual modality agents, with the results recognized and passed to a facilitator. These results, typically an *n*-best list of conjectured lexical items and related time-stamp information, then are routed to appropriate agents for further language processing. Next, sets of meaning fragments derived from the speech and pen signals arrive at the multimodal integrator. This agent decides whether and how long to wait for recognition results from other modalities, based on the system's temporal thresholds. It fuses the meaning fragments into a semantically- and temporally-compatible whole interpretation before passing the results back to the facilitator. At this point, the system's final multimodal interpretation is confirmed by the interface, delivered as multimedia feedback to the user, and executed by any relevant applications. In summary, multi-agent architectures provide essential infrastructure for coordinating the many complex modules needed to implement multimodal system processing, and they permit doing so in a distributed manner that is compatible with the trend toward mobile computing.

The core of multimodal systems based on semantic fusion involves algorithms that integrate common meaning representations derived from speech, gesture, and other modalities into a combined final interpretation. The semantic fusion operation requires a common meaning-representation framework for all modalities, and a well-defined operation for combining partial meanings that arrive from different signals. To fuse information from different modalities, various research groups have independently converged on a strategy of recursively matching and merging attribute/value data structures, although using a variety of different algorithms (Vo & Wood, 1996; Cheyer & Julia, 1995; Pavlovic & Huang, 1998; Shaikh et al., 1997). This approach is considered a *frame-based integration* technique. An alternative logic-based approach derived from computational linguistics (Carpenter, 1990, 1992; Calder, 1987) involves the use of *typed feature structures* and *unification-based integration*, which is a more general and well-understood approach. Unification-based integration techniques also have been applied to multimodal system design (Cohen et al., 1997; Johnston et al., 1997; Wu et al., 1999). Feature-structure unification is considered well suited to multimodal integration, because unification can combine complementary or redundant input from both modes, but it rules out contradictory input. Given this foundation for multimodal integration, more research still is needed on the development of canonical meaning representations that are common among different input modes which will need to be represented in new types of multimodal systems.

When statistical processing techniques are combined with a symbolic unification-based approach that merges feature structures, then the multimodal architecture that results is a *hybrid symbolic/statistical* one. Hybrid architectures represent one major new direction for multimodal system development. Multimodal architectures also can be hybrids in the sense of combining Hidden Markov Models (HMMs) and Neural Networks (NNs). New hybrid architectures potentially are capable of achieving very robust functioning, compared with either an early- or late-fusion approach alone. For example, the Members-Teams-Committee (MTC) hierarchical recognition technique, which is a hybrid symbolic/statistical multimodal integration framework trained over a labeled multimodal corpus, recently achieved 95.26% correct recognition performance, or within 1.4% of the theoretical system upper bound (Wu et al., 1999). Other architectural approaches and contributions to processing multimodal information have been summarized elsewhere (Oliver & Horvitz, 2005; Potamianos et al., 2003).

WHAT ARE THE MAIN FUTURE DIRECTIONS FOR MULTIMODAL INTERFACE DESIGN?

The computer science community is just beginning to understand how to design innovative, well integrated, and robust multimodal systems. To date, most multimodal systems remain bimodal, and recognition technologies related to several human senses (e.g., haptics, smell, taste) have yet to be well represented within multimodal interfaces. The design and development of new types of systems that include such modes will not be achievable through intuition. Rather, it will depend on knowledge of the natural integration patterns that typify people's combined use of various input modes. This means that the successful design of multimodal systems will continue to require guidance from cognitive science on the coordinated human perception and production of natural modalities. In this respect, multimodal systems only can flourish through multidisciplinary cooperation, as well as teamwork among those representing expertise in the component technologies.

Most of the systems outlined in this chapter have been built during the past 15 years, and they are research-level systems. However, in some cases they have developed well beyond the prototype stage, and are being integrated with other software at academic and federal sites, or appearing as newly shipped products. To achieve wider commercialization of multimodal interfaces, such systems will need to develop more powerful and

general methods of natural language and dialogue processing, and temporal modeling and processing of incoming signals. In addition, multimodal datasets and tools are very much needed to build applications more rapidly in a wide range of domains, including for newly emerging collaborative multimodal applications such as meeting support and education (Barthelmess, Kaiser, Huang, & Demirdjian, 2005; Cohen & McGee, 2004; Danninger et al., 2005; Gatica-Perez, Lathoud, Odobez, & McCowan, 2005; McGee, 2003; Pentland, 2005). The many mobile multimodal interfaces currently being built also will require active adaptation to the user, task, ongoing dialogue, and environmental context, which is another very active area of recent work (Gorniak & Roy, 2005; Gupta, 2004; Huang & Oviatt, 2005; Jain & Ross, 2002; Potamianos et al., 2003; Xiao et al., 2003). To facilitate the speed and generality of multimodal interface adaptation to these important variables, future work will need to integrate new machine learning techniques that are now being developed to handle asynchronous and heterogeneous data (Bengio, 2004; McCowan et al., 2005). Finally, in the future a coherent theoretical framework needs to be developed to account for multimodal interaction patterns. This will be invaluable for proactively guiding the design of new multimodal interfaces to be compatible with human capabilities and limitations. Current work in cognitive neuroscience and multisensory perception are beginning to provide an empirical and theoretical basis for this future interface design (Calvert et al., 2004; Ernst & Bulthoff, 2004).

In conclusion, multimodal interfaces are just beginning to model human-like sensory perception. They are recognizing and identifying actions, language, and people that have been seen, heard, or in other ways experienced in the past. They literally reflect and acknowledge the existence of human users, empower them in new ways, and create for them a "voice." They also can be playful and self-reflective interfaces that suggest new forms of human identity as we interact face to face with animated personas representing our own kind. In all of these ways novel multimodal interfaces, as primitive as their early bimodal instantiations may be, represent a new multidisciplinary science, a new art form, and a socio-political statement about our collective desire to humanize the technology we create.

ACKNOWLEDGMENTS

I'd like to thank the National Science Foundation for their support over the past decade, which has enabled me to pursue basic exploratory research on many aspects of multimodal interaction, interface design, and system development. The preparation of this chapter has been supported by NSF Grants IRI-9530666, IIS-0117868, and NSF Special Extension for Creativity (SEC) Grant IIS-9530666. This work also has been supported by contracts DABT63-95-C-007, N66001-99-D-8503, and NBCHD030010 from DARPA's Information Technology and Information Systems Office, and Grant No. N00014-99-1-0377 from ONR. Any opinions, findings or conclusions expressed in this chapter are those of the author and do not necessarily reflect the views of the federal agencies sponsoring this work.

I'd also like to thank Phil Cohen and others in the Center for Human-Computer Communication for many insightful discussions, and Dana Director and Rachel Coulston for expert assistance with manuscript preparation. Finally, I wish to acknowledge LEA, Inc. for giving permission to reprint Figs. 21.1 and 21.5, and to acknowledge ACM for allowing the reprint of Figs. 21.3 and 21.4.

References

Abry, C., Lallouache, M.-T., & Cathiard, M.-A. (1996). How can coarticulation models account for speech sensitivity to audio-visual desynchronization? In D. G. Stork & M. E. Hennecke (Eds.), *Speechreading by Humans and Machines: Models, Systems and Applications* (pp. 247–255). New York: Springer Verlag.

Adjoudani, A., & Benoit, C. (1995). Audio-visual speech recognition compared across two architectures. *Proc. of the Eurospeech Conference Vol. 2* (pp. 1563–1566). Madrid, Spain.

Almor, A. (1999). Noun-phrase anaphora and focus: The informational load hypothesis. *Psychological Review, 106*, 748–765.

Argyle, M. (1972). nonverbal communication in human social interaction. In R. Hinde (Ed.), *Nonverbal Communication* (pp. 243–267). Cambridge, UK: Cambridge University Press.

Baddeley, A. (1992). Working memory. *Science, 255*, 556–559.

Bangalore, S., & Johnston, M. (2000). Integrating multimodal language processing with speech recognition. In B. Yuan, T. Huang & X. Tang (Eds.), *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP'2000*) *Vol. 2* (pp. 126–129). Beijing: Chinese Friendship Publishers.

Barthelmess, P., Kaiser, E., Huang, X., & Demirdjian, D. (2005). Distributed pointing for multimodal collaboration over sketched diagrams, *Proceedings of the Seventh International Conference on Multimodal Interfaces* (pp. 10–17) New York, N.Y.: ACM Press.

Bengio, S. (2004). Multimodal speech processing using asynchronous Hidden Markov Models, *Information Fusion, 5*(2), 81–89.

Benoit, C., Guiard-Marigny, T., Le Goff, B., & Adjoudani, A. (1996). Which components of the face do humans and machines best speechread? In D. G. Stork & M. E. Hennecke, (Eds.), *Speechreading by Humans and Machines: Models, Systems, and Applications: Vol. 150 of NATO ASI Series. Series F: Computer and Systems Sciences* (pp. 315–325). Berlin, Germany: Springler-Verlag.

Benoit, C., & Le Goff, B. (1998). Audio-visual speech synthesis from French text: Eight years of models, designs and evaluation at the ICP. *Speech Communication, 26*, 117–129.

Benoit, C., Martin, J.-C., Pelachaud, C., Schomaker, L., & Suhm, B. (2000). Audio-visual and multimodal speech-based systems. In D. Gibbon, I. Mertins, & R. Moore (Eds.), *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation* (pp. 102–203). Amsterdam, the Netherlands: Kluwer.

Bernstein, L., & Benoit, C. (1996). For speech perception by humans or machines, three senses are better than one. *Proceedings of the International Conference on Spoken Language Processing,* (*ICSLP 96*) *Vol. 3* (pp. 1477–1480). New York: IEEE press.

Bers, J., Miller, S., & Makhoul, J. (1998). Designing conversational interfaces with multimodal interaction. *DARPA Workshop on Broadcast News Understanding Systems*, 319–321.

Bolt, R. A. (1980). Put-that-there: Voice and gesture at the graphics interface. *Computer Graphics, 14*(3), 262–270.

Bregler, C., & Konig, Y. (1994). Eigenlips for robust speech recognition. *Proceedings of the International Conference on Acoustics Speech and Signal Processing* (*IEEE-ICASSP*) *Vol. 2*, 669–672.

Bregler, C., Manke, S., Hild, H., & Waibel, A. (1993). Improving connected letter recognition by lipreading. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (*IEEE-ICASSP*) *Vol.1*, 557–560.

Brooke, N. M., & Petajan, E. D. (1986). Seeing speech: Investigations into the synthesis and recognition of visible speech movements using automatic image processing and computer graphics. *Proceedings International Conference Speech Input and Output: Techniques and Applications, 258*, 104–109.

Calder, J. (1987). Typed unification for natural language processing. In E. Klein & J. van Benthem (Eds.), *Categories, Polymorphisms, and Unification* (pp. 65–72). Edinburgh, Scotland: Center for Cognitive Science, University of Edinburgh.

Calvert, G., Spence, C., & Stein, B. E. (Eds.). (2004). *The Handbook of Multisensory Processing*. Cambridge, MA: MIT Press.

Carpenter, R. (1990). Typed feature structures: Inheritance, (in)equality, and extensionality. *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing*, 9–18. Tilburg the Netherlands: Institute for Language Technology and Artificial Intelligence, Tilburg University.

Carpenter, R. (1992). *The logic of typed feature structures.* Cambridge, England: Cambridge University Press.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. (Eds.). (2000). *Embodied conversational agents.* Cambridge, MA: MIT Press.

Cheyer, A. (1998, January). MVIEWS: Multimodal tools for the video analyst. *International Conference on Intelligent User Interfaces* (*IUI'98*), (pp. 55–62). New York: ACM Press.

Cheyer, A., & Julia, L. (1995, May). Multimodal maps: An agent-based approach. *International Conference on Cooperative Multimodal Communication,* (*CMC'95*), (pp. 103–113). Eindhoven, The Netherlands.

Choudhury, T., Clarkson, B., Jebara, T., & Pentland, S. (1999). Multimodal person recognition using unconstrained audio and video. *Proceedings of the 2nd International Conference on Audio-and-Video-based Biometric Person Authentication* (pp. 176–181). Washington, DC:.

Codella, C., Jalili, R., Koved, L., Lewis, J., Ling, D., Lipscomb, J., et al. (1992). Interactive simulation in a multi-person virtual world. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'92*) (pp. 329–334). New York: ACM Press.

Cohen, M. M., & Massaro, D. W. (1993). Modeling coarticulation in synthetic visual speech. In M. Magnenat-Thalmann & D. Thalmann (Eds.), *Models and Techniques in Computer Animation.* Tokyo: Springer-Verlag.

Cohen, P. R., Cheyer, A., Wang, M., & Baeg, S. C. (1994). An open agent architecture. *AAAI '94 Spring Symposium Series on Software Agents*, 1–8. (Reprinted in Huhns and Singh (Eds.). (1997). *Readings in Agents* (pp. 197–204). San Francisco: Morgan Kaufmann.)

Cohen, P. R., Dalrymple, M., Moran, D. B., Pereira, F. C. N., Sullivan, J. W., Gargan, R. A., et al. (1989). Synergistic use of direct manipulation and natural language. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'89*), (pp. 227–234). New York: ACM Press. Reprinted in Maybury & Wahlster (Eds.), (1998). *Readings in Intelligent User Interfaces* (pp. 29–37). San Francisco: Morgan Kaufmann.)

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., et al. (1997). Quickset: Multimodal interaction for distributed applications. *Proceedings of the Fifth ACM International Multimedia Conference* (pp. 31–40). New York: ACM Press.

Cohen, P. R., & McGee, D. (2004, January). Tangible multimodal interfaces for safety-critical applications, *Communications of the ACM, Vol. 47*. (pp. 41–46). New York, NY: ACM Press.

Cohen, P. R., McGee, D. R., & Clow, J. (2000). The efficiency of multimodal interaction for a map-based task. *Proceedings of the Language Technology Joint Conference* (*ANLP-NAACL 2000*) (pp. 331–338). Seattle, WA: Association for Computational Linguistics Press. Cohen, P. R., & Oviatt, S. L. (1995). The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences, 92*(22), (pp. 9921–9927). Washington, DC: National Academy of Sciences Press.

Condon, W. S. (1988). An analysis of behavioral organization. *Sign Language Studies, 58*, 55–88.

Dahlbäck, N., Jëonsson, A., & Ahrenberg, L. (1992, January). Wizard of Oz studies—why and how. In W. D. Gray, W. E. Hefley, & D. Murray (Eds.), *Proceedings of the International Workshop on Intelligent User Interfaces* (pp. 193–200). New York: ACM Press.

Danninger, M., Flaherty, G., Bernardin, K., Ekenel, H., Kohler, T., Malkin, R., et al. (2005). The connector- Facilitaing context-aware communication, *Proceedings of the International Conference on Multimodal Interfaces* (pp. 69–75) New York: ACM.

Denecke, M., & Yang, J. (2000). Partial Information in Multimodal Dialogue. *Proceedings of the International Conference on Multimodal Interaction*, (pp. 624–633). Beijing, China:Friendship Press.

Duncan, L., Brown, W., Esposito, C., Holmback, H., & Xue, P. (1999). Enhancing virtual maintenance environments with speech understanding. Boeing M&CT TechNet.

Dupont, S., & Luettin, J. (2000, September). Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia, 2*(3), 141–151. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Ekman, P. (1992, January). Facial expressions of emotion: New findings, new questions. *American Psychological Society, 3*(1), 34–38.

Ekman, P., & Friesen, W. (1978). *Facial Action Coding System.* Mountain View, CA: Consulting Psychologists Press.

Encarnacao, L. M., & Hettinger, L. (Eds.) (2003, September/October) Perceptual Multimodal Interfaces (special issue), *IEEE Computer Graphics and Applications*. Mountain View, CA.

Epps, J., Oviatt, S., & Chen, F. (2004). Integration of speech and gesture input during multimodal interaction. *Proceedings of the Australian International Conference on Computer-Human Interaction* (*OzCHI*).

Ernst, M., & Bulthoff, H. (2004, April). Merging the sense into a robust whole percept. *Trends in Cognitive Science, 8*(4), 162–169.

Fell, H., Delta, H., Peterson, R., Ferrier, L., Mooraj, Z., & Valleau, M. (1994). Using the baby-babble-blanket for infants with motor problems. *Proceedings of the Conference on Assistive Technologies* (*ASSETS'94*), 77–84.

Flanagan, J., & Huang, T. (Eds.). (2003, September). Multimodal Human Computer Interfaces (special issue), *Proceedings of IEEE, Vol. 91* (9).

Fridlund, A. (1994). *Human facial expression: An evolutionary view.* New York: Academic Press.

Fukumoto, M., Suenaga, Y., & Mase, K. (1994). Finger-pointer: Pointing interface by image processing. *Computer Graphics, 18*(5), 633–642.

Fuster-Duran, A. (1996). Perception of conflicting audio-visual speech: an examination across Spanish and German, In D. G. Stork & M. E. Hennecke (Eds.), *Speechreading by Humans and Machines: Models, Systems and Applications* (pp. 135–143). New York: Springer Verlag.

Gatica-Perez, D., Lathoud, G., Odobez, J.-M., & McCowan, I. (2005). Multimodal multispeaker probabilistic tracking in meetings. *Proceedings of the Seventh International Conference on Multimodal Interfaces*, (pp. 183–190). New York: ACM.

Gorniak, P., & Roy, D. (2005). Probabilistic grounding of situated speech using plan recognition and reference resolution. *Proceedings of the Seventh International Conference on Multimodal Interfaces* (pp. 138–143). New York: ACM.

Gupta, A. (2004). Dynamic time windows for multimodal input fusion, *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP'04*). 1009–1012

Hadar, U., Steiner, T. J., Grant, E. C., & Rose, C. F. (1983). Kinematics of head movements accompanying speech during conversation. *Human Movement Science, 2*, 35–46.

Hauptmann, A. G. (1989). Speech and gestures for graphic image manipulation. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'89*), *Vol. 1* (pp. 241–245). New York: ACM Press.

Holzman, T. G. (1999). Computer-human interface solutions for emergency medical care. *Interactions, 6*(3), 13–24.

Huang, X., Acero, A., Chelba, C., Deng, L., Duchene, D., Goodman, J., et al. (2000). MiPad: A next-generation PDA prototype. *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP 2000*) *Vol. 3* (pp. 33–36). Beijing, China: Chinese Military Friendship Publishers.

Huang, X. & Oviatt, S. Toward adaptive information fusion in multimodal systems, *Second Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MIML'05)*, Springer Lecture Notes in Computer Science, ed. by Steve Renals & Samy Bengio, Springer-Verlag GmbH, 2006, vol. 3869, 15–27.

Iyengar, G., Nock, H., & Neti, C. (2003). Audio-visual synchrony for detection of monologues in video archives. *Proceedings of ICASSP*.

Jain, A., Hong, L., & Kulkarni, Y. (1999). A multimodal biometric system using fingerprint, face and speech, *2nd International Conference on Audio- and Video-based Biometric Person Authentication* (pp. 182–187). Washington DC:

Jain, A., & Ross, A. (2002). Learning user-specific parameters in a multibiometric system. *Proceedings of the International Conference on Image Processing* (*ICIP*), September 22–25, 2002. (pp. 57–60) Rochester, NY:

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., & Smith, I. (1997). Unification-based multimodal integration. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (pp. 281–288). San Francisco: Morgan Kaufmann. Karshmer, A. I., & Blattner, M. (editors). (1998). *Proceedings of the 3rd International ACM Proceedings of the Conference on Assistive Technologies* (*ASSETS'98*). (URL http://www.acm.org/sigcaph/assets/assets98/assets98index.html).

Kendon, A. (1980). Gesticulation and speech: Two aspects of the process of utterance. In M. Key (Ed.), *The Relationship of Verbal and Nonverbal Communication* (pp. 207–227). The Hague, Netherlands: Mouton.

Kobsa, A., Allgayer, J., Reddig, C., Reithinger, N., Schmauks, D., Harbusch, K., et al. (1986). Combining Deictic Gestures and Natural Language for Referent Identification. *Proceedings of the 11th International Conference on Computational Linguistics* (pp. 356–361). Bonn, Germany Proceedings of COLING '86, August 25–29, 1986, University of Bonn, Germany. Published by Institut für angewandte Kommunikations-und Sprachforschung e.V. (IKS).

Kopp, S., Tepper, P., & Cassell, J. (2004). Towards integrated microplanning of language and iconic gesture for multimodal output. *Proceedings of the 6th International Conference on Multimodal Interfaces*, (pp. 97–104). New York: ACM.

Koons, D., Sparrell, C., & Thorisson, K. (1993). Integrating simultaneous input from speech, gaze, and hand gestures. In M. Maybury (Ed.), *Intelligent Multimedia Interfaces* (pp. 257–276). Cambridge, MA: MIT Press.

Kricos, P. B. (1996). Differences in visual intelligibility across talkers. In D. G. Stork & M. E. Hennecke (Eds.), *Speechreading by Humans and Machines: Models, Systems and Applications* (pp. 43–53). New York: Springer Verlag.

Kumar, S., & Cohen, P. R. (2000). Towards a fault-tolerant multi-agent system architecture. *Fourth International Conference on Autonomous Agents 2000* (pp. 459–466). Barcelona, Spain: ACM Press.

Leatherby, J. H., & Pausch, R. (1992, July). Voice input as a replacement for keyboard accelerators in a mouse-based graphical editor: An empirical study. *Journal of the American Voice Input/Output Society, 11*(2).

Martin, D. L., Cheyer, A. J., & Moran, D. B. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence, 13*, 91–128.

Massaro, D. W. (1996). Bimodal speech perception: A progress report. In D. G. Stork & M. E. Hennecke (Eds.), *Speechreading by Humans and Machines: Models, Systems and Applications* (pp. 79–101). New York: Springer Verlag.

Massaro, D. W., & Cohen, M. M. (1990, January). Perception of synthesized audible and visible speech. *Psychological Science, 1*(1), 55–63.

Massaro, D. W., & Stork, D. G. (1998). Sensory integration and speechreading by humans and machines. *American Scientist, 86*, 236–244.

McCowan, I., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., & Zhang, D. (2005). Automatic analysis of multimodal group actions in meetings, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), *27*(3), 305–317.

McGee, D. (2003, June). *Augmenting environments with multimodal interaction*. Unpublished manuscript, Oregon Health & Science University.

McGrath, M., & Summerfield, Q. (1985). Intermodal timing relations and audio-visual speech recognition by normal-hearing adults. *Journal of the Acoustical Society of America, 77*(2), 678–685.

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices, *Nature, 264*, 746–748.

McLeod, A., & Summerfield, Q. (1987). Quantifying the contribution of vision to speech perception in noise. *British Journal of Audiology, 21*, 131–141.

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought.* Chicago: University of Chicago Press.

Meier, U., Hürst, W., & Duchnowski, P. (1996). Adaptive bimodal sensor fusion for automatic speechreading. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (*IEEE-ICASSP*) (pp. 833–836). Los Alamitos, CA:IEEE Press

Morency, L.-P., Sidner, C., Lee, C., & Darrell, T. (2005). Contextual recognition of head gestures. *Proceedings of the Seventh International Conference on Multimodal Interfaces* (pp. 18–24). New York: ACM.

Morimoto, C., Koons, D., Amir, A., Flickner, M., & Zhai, S. (1999). Keeping an Eye for HCI. *Proceedings of SIBGRAPI'99, XII Brazilian Symposium on Computer Graphics and Image Processing*, 171–176.

Mousavi, S. Y., Low, R., & Sweller, J. (1995). Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of Educational Psychology, 87*(2), 319–334.

Naughton, K. (1996). Spontaneous gesture and sign: A study of ASL signs co-occurring with speech. In L. Messing (Ed.), *Proceedings of the Workshop on the Integration of Gesture in Language & Speech* (pp. 125–134). Newark, DE:University of Delaware

Neal, J. G., & Shapiro, S. C. (1991). Intelligent multimedia interface technology. In J. Sullivan & S. Tyler (Eds.), *Intelligent User Interfaces* (pp.11–43). New York: ACM Press.

Negroponte, N. (1978, December). *The Media Room.* Cambridge, MA: MIT Press.

Neti, C., Iyengar, G., Potamianos, G., & Senior, A. (2000). Perceptual interfaces for information interaction: Joint processing of audio and visual information for human-computer interaction. In B. Yuan, T. Huang & X. Tang (Eds.), *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP'2000*), *Vol. 3* (pp. 11–14). Beijing, China: Chinese Friendship Publishers.

Oliver, N., & Horvitz, E. (2005). S-SEER: Selective perception in a multimodal office activity system. *International Journal of Computer Vision and Image Understanding*.

Oviatt, S. L. (1997). Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction* [*Special issue on Multimodal Interfaces*], *12*, 93–129.

Oviatt, S. L. (1999a). Mutual disambiguation of recognition errors in a multimodal architecture. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'99*) (pp. 576–583). New York: ACM Press.

Oviatt, S. L. (1999b). Ten myths of multimodal interaction. *Communications of the ACM, 42*(11), 74–81.

Oviatt, S. L. (2000a). Multimodal system processing in mobile environments. *Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software Technology* (*UIST'2000*) (pp. 21–30). New York: ACM Press.

Oviatt, S. L. (2000b). Taming recognition errors with a multimodal architecture. *Communications of the ACM, 43*(9), 45–51.

Oviatt, S. L. (2000c). Multimodal signal processing in naturalistic noisy environments. In B. Yuan, T. Huang, & X. Tang (Eds.), *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP'2000*)*, Vol. 2* (pp. 696–699). Beijing, China: Chinese Friendship Publishers.

Oviatt, S. L. (2002). Breaking the robustness barrier: Recent progress in the design of robust multimodal systems. *Advances in Computers 56*, 305–341.

Oviatt, S. L. (2003, September/October). Advances in robust multimodal interfaces, *IEEE Computer Graphics and Applications,* 62–68.

Oviatt, S. L., Bernard, J., & Levow, G. (1999). Linguistic adaptation during error resolution with spoken and multimodal systems. *Language and Speech, 41*(3–4), 415–438.

Oviatt, S. L., & Cohen, P. R. (1991). Discourse structure and performance efficiency in interactive and noninteractive spoken modalities. *Computer Speech and Language, 5*(4), 297–326.

Oviatt, S. L., & Cohen, P. R. (2000, March). Multimodal systems that process what comes naturally. *Communications of the ACM, 43*(3), 45–53.

Oviatt, S. L., Cohen, P. R., Fong, M. W., & Frank, M. P. (1992). A rapid semi-automatic simulation technique for investigating interactive speech and handwriting. *Proceedings of the International Conference on Spoken Language Processing, 2*, 1351–1354.

Oviatt, S. L., Cohen, P. R., & Wang, M. Q. (1994). Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. *Speech Communication, 15*, 283–300.

Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., et al. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction, 15*(4), 263–322.

Oviatt, S. L., Coulston, R., & Lunsford, R. (2004). When do we interact multimodally? Cognitive load and multimodal communication patterns, *Proceedings of the Sixth International Conference on Multimodal Interfaces* (*ICMI'04*) 129–136.

Oviatt, S. L., Coulston, R., Shriver, S., Xiao, B., Wesson, R., Lunsford, R., et al. (2003). Toward a theory of organized multimodal integration patterns during human-computer interaction, *Proceedings of the International Conference on Multimodal Interfaces* (*ICMI'03*) (pp. 44–51). New York: ACM Press.

Oviatt, S. L., DeAngeli, A., & Kuhn, K. (1997). Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of Conference on Human Factors in Computing Systems* (*CHI'97*) (pp. 415–422). New York: ACM Press.

Oviatt, S. L, Flickner, M., & Darrell, T. (Eds.) (2004, January). Multimodal interfaces that flex, adapt and persist, *Communications of the ACM 47*(1), 30–33.

Oviatt, S. L., & Kuhn, K. (1998). Referential features and linguistic indirection in multimodal language. *Proceedings of the International Conference on Spoken Language Processing, 6*, (pp. 2339–2342). Syndey, Australia: ASSTA, Inc.

Oviatt, S. L. & Lunsford, R. (2005). Multimodal interfaces for cell phones and mobile technology, *International Journal of Speech Technology, 8*(2), Springer & Springer-link on-line, 127–132

Oviatt, S. L., Lunsford, R., & Coulston, R. (2005). Individual differences in multimodal integration patterns: What are they and why do they exist? *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'05*)*, CHI Letters* (pp. 241–249). New York: ACM Press.

Oviatt, S. L., & Olsen, E. (1994). Integration themes in multimodal human-computer interaction. In Shirai, Furui, & Kakehi (Eds.), *Proceedings of the International Conference on Spoken Language Processing, 2*, 551–554.

Oviatt, S. L., & van Gent, R. (1996). Error resolution during multimodal human-computer interaction. *Proceedings of the International Conference on Spoken Language Processing, 2*, 204–207.

Pankanti, S., Bolle, R.M., & Jain, A. (Eds.). (2000). Biometrics: The future of identification. *Computer, 33*(2), 46–80.

Pavlovic, V., Berry, G., & Huang, T. S. (1997). Integration of audio/visual information for use in human-computer intelligent interaction. *Proceedings of IEEE International Conference on Image Processing,* 121–124.

Pavlovic, V., & Huang, T. S. (1998). Multimodal prediction and classification on audio-visual features. *AAAI'98 Workshop on Representations for Multi-modal Human-Computer Interaction* (pp. 55–59). Menlo Park, CA: AAAI Press.

Pavlovic, V., Sharma, R., & Huang, T. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7) 677–695.

Pentland, S. (2005, March). Socially aware computation and communication, *IEEE Computer*, 63–70.

Potamianos, G., & Neti, C. (2001). Automatic speechreading of impaired speech, *Proceedings of the International Conference on Auditory-Visual Speech Processing*, 177–182.

Potamianos, G., Neti, C., Gravier, G., & Garg, A. (2003, September). Automatic recognition of audio-visual speech: Recent progress and challenges, *Proceedings of the IEEE, Vol. 91* (9). September, 2003.

Petajan, E. D. (1984). *Automatic Lipreading to Enhance Speech Recognition*. Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign.

Pfleger, N. (2004). Context-based multimodal fusion. *Proceedings of the 6th International Conference on Multimodal Interfaces* (pp. 265–272). New York: ACM.

Poddar, I., Sethi, Y., Ozyildiz, E., & Sharma, R. (1998, November). Toward natural gesture/speech HCI: A case study of weather narration. In M. Turk, (Ed.), *Proceedings 1998 Workshop on Perceptual User Interfaces* (*PUI'98*) (pp. 1–6). San Francisco, CA.

Reithinger, N., Alexandersson, J., Becker, T., Blocher, A., Engel, R., Lockelt, M., et al. (2003). Multimodal architectures and frameworks: SmartKom: adaptive and flexible multimodal access to multiple applications, *Proceedings of the 5th International Conference on Multimodal Interfaces* (pp. 101–108). New York: ACM.

Robert-Ribes, J., Schwartz, J.-L., Lallouache, T., & Escudier, P. (1998). Complementarity and synergy in bimodal speech: Auditory, visual, and auditory-visual identification of French oral vowels in noise. *Journal of the Acoustical Society of America, 103*(6) 3677–3689.

Rogozan, A., & Deglise, P. (1998). Adaptive fusion of acoustic and visual sources for automatic speech recognition. *Speech Communication, 26*(1–2), 149–161.

Rubin, P., Vatikiotis-Bateson, E., & Benoit, C. (Eds.). (1998). Audio-visual speech processing *Speech Communication, 26*, 1–2.

Rudnicky, A., & Hauptman, A. (1992). Multimodal interactions in speech systems. In M. Blattner & R. Dannenberg (Eds.), *Multimedia Interface Design* (pp. 147–172). New York: ACM Press.

Sekiyama, K., & Tohkura, Y. (1991). McGurk effect in non-English listeners: Few visual effects for Japanese subjects hearing Japanese syllables of high auditory intelligibility. *Journal of the Acoustical Society of America, 90*, 1797–1805.

Seneff, S., Goddeau, D., Pao, C., & Polifroni, J. (1996). Multimodal discourse modelling in a multi-user multi-domain environment. In T. Bunnell & W. Idsardi (Eds.), *Proceedings of the International Conference on Spoken Language Processing, Vol. 1* (pp. 192–195).

Shaikh, A., Juth, S., Medl, A., Marsic, I., Kulikowski, C., & Flanagan, J. (1997). An architecture for multimodal information fusion. *Proceedings of the Workshop on Perceptual User Interfaces* (*PUI'97*), 91–93.

Sharma, R., Huang, T. S., Pavlovic, V. I., Schulten, K., Dalke, A., Phillips, J., et al. (1996, August). Speech/gesture interface to a visual computing environment for molecular biologists. *Proceedings of 13th International Conference on Pattern Recognition* (*ICPR 96*), *Vol. 3*, 964–968.

Sharma, R., Pavlovic, V. I., & Huang, T. S. (1998). Toward multimodal human-computer interface. *Proceedings IEEE, 86*(5), 853–860.

Silsbee, P. L., & Su, Q. (1996). Audiovisual sensory intergration using Hidden Markov Models. In D. G. Stork & M. E. Hennecke (Eds.), *Speechreading by Humans and Machines: Models, Systems and Applications* (pp. 489–504). New York: Springer Verlag.

Siroux, J., Guyomard, M., Multon, F., & Remondeau, C. (1995, May). Modeling and processing of the oral and tactile activities in the Georal tactile system. *International Conference on Cooperative Multimodal Communication, Theory & Applications*. Eindhoven, Netherlands.

Stork, D. G., & Hennecke, M. E. (Eds.). (1995). *Speechreading by Humans and Machines.* New York: Springer Verlag.

Suhm, B. (1998). *Multimodal interactive error recovery for non-conversational speech user interfaces.* Doctoral thesis. Fredericiana University, Germany: Shaker Verlag.

Sumby, W. H., & Pollack, I. (1954). Visual contribution to speech intelligibility in noise. *Journal of the Acoustical Society of America, 26*, 212–215.

Summerfield, A. Q. (1992). Lipreading and audio-visual speech perception, *Philosophical Transactions of the Royal Society of London, Series B, 335*, 71–78.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*, 257–285.

Tang, A., McLachlan, P., Lowe, K., Saka, C., & MacLean, K. (2005). Perceiving ordinal data haptically under workload, *Proceedings of the Seventh International Conference on Multimodal Interfaces*, (pp. 317–324). New York: ACM.

Tomlinson, M. J., Russell, M. J., & Brooke, N. M. (1996). Integrating audio and visual information to provide highly robust speech recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (*IEEE-ICASSP*) *Vol. 2*, 821–824.

Turk, M., & Robertson, G. (Eds.). (2000). Perceptual user interfaces. *Communications of the ACM, 43*(3), 32–70.

Vatikiotis-Bateson, E., Munhall, K. G., Hirayama, M., Lee, Y. V., & Terzopoulos, D. (1996). The dynamics of audiovisual behavior of speech. In D. G. Stork and M. E. Hennecke, (Eds.), *Speechreading by Humans and Machines: Models, Systems, and Applications, Vol. 150 of NATO ASI Series. Series F: Computer and Systems Sciences* (pp. 221–232). Berlin, Germany: Springler-Verlag.

Vo, M. T., & Wood, C. (1996). Building an application framework for speech and pen input integration in multimodal learning interfaces. *Proceedings of the International Conference on Acoustics Speech and Signal Processing* (*IEEE-ICASSP*) *Vol. 6*, 3545–3548.

Vo, M. T., Houghton, R., Yang, J., Bub, U., Meier, U., Waibel, A., et al. (1995). Multimodal learning interfaces. *Proceedings of the DARPA Spoken Language Technology Workshop.*

Wahlster, W. (1991). User and discourse models for multimodal communciation. In J. W. Sullivan & S. W. Tyler (Eds.), *Intelligent User Interfaces* (pp. 45–67). New York: ACM Press.

Wahlster, W. (2001, March). SmartKom: multimodal dialogs with mobile web users. *Proceedings of the Cyber Assist International Symposium*, 33–34.

Waibel, A., Suhm, B., Vo, M. T., & Yang, J. (1997, April). Multimodal interfaces for multimedia information agents. *Proceedings of the International Conference on Acoustics Speech and Signal Processing* (*IEEE-ICASSP*) *Vol. 1*, 167–170.

Wang, J. (1995). Integration of eye-gaze, voice and manual response in multimodal user interfaces. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* 3938–3942.

Wickens, C. D., Sandry, D. L., & Vidulich, M. (1983). Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors, 25*, 227–248.

Wu, L., Oviatt, S., &. Cohen, P. (1999). Multimodal integration—A statistical view. *IEEE Transactions on Multimedia, 1*(4), 334–341.

Xiao, B., Lunsford, R., Coulston, R., Wesson, R., & Oviatt, S. L. (2003). Modeling multimodal integration patterns and performance in seniors: Toward adaptive processing of individual differences, *Proceedings of the International Conference on Multimodal Interfaces* (*ICMI'03*) (pp. 265–272). New York: ACM Press.

Xiao, B., Girand, C., & Oviatt, S. (2002, September). Multimodal integration patterns in children. *Proceedings of the International Conference on Spoken Language Processing* (*ICSLP'02*), 629–632.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI'99*) (pp. 246–253). New York: ACM Press.

·22·

ADAPTIVE INTERFACES AND AGENTS

Anthony Jameson
German Research Center for Artificial Intelligence (DFKI)
and International University in Germany

Introduction **434**
Concepts 434
Functions: Supporting System Use **435**
Taking Over Parts of Routine Tasks 435
Adapting the Interface 436
Helping With System Use 437
Mediating Interaction with the Real World 438
Controlling a Dialogue 439
Functions: Supporting Information Acquisition **440**
Helping Users to Find Information 440
Support for browsing 441
Support for query-based search or filtering 442
Spontaneous Provision of Information 442
Recommending Products 442
Tailoring Information Presentation 443
Supporting Collaboration 444
Supporting Learning 445
Usability Challenges **446**
Threats to Predictability and Comprehensibility 447
Threats to Controllability 447
Obtrusiveness 448
Threats to Privacy 448
Breadth of Experience 448
Dealing With Trade-offs 448
Obtaining Information About Users **449**
Explicit Self-Reports and -Assessments 449
Self-reports about objective personal characteristics 449
Self-assessments of interests and knowledge 449
Self-reports on specific evaluations 450
Responses to test items 450
Nonexplicit Input 450
Naturally Occurring Actions 450
Previously Stored Information 450
Low-Level Indices of Psychological States 451
Signals Concerning the Current Surroundings 451
Special Considerations Concerning Empirical Methods **451**
Use of Data Collected With a Nonadaptive System 451
Early Studies of Usage Scenarios and User Requirements 452
Wizard of Oz Studies 452
Comparisons With the Work of Human Designers 453
Experimental Comparisons of Adaptive and Nonadaptive Systems 453
Taking Into Account Individual Differences 454
Checking Usability Under Realistic Conditions 454
The Future of User-Adaptive Systems **454**
Growing Need for User-Adaptivity 454
Diversity of users and contexts of use 454
Number and complexity of interactive systems . . . 454
Scope of information to be dealt with 454
Increasing Feasibility of Successful Adaptation 455
Ways of acquiring information about users 455
Advances in techniques for learning, inference, and decision 455
Attention to empirical methods 455
Acknowledgments **455**
References **455**

INTRODUCTION

This chapter covers a broad range of interactive systems. They all have one idea in common: It can be worthwhile for a system to learn something about each individual user and adapt its behavior to them in some nontrivial way.

An example familiar to most readers is shown in Fig. 22.1. A visitor to Amazon.com has just explicitly requested recommendations, without having specified a particular type of product. During the user's past visits, Amazon.com has learned something about his or her interests, on the basis of items he or she has purchased and ratings he or she has made. Therefore, the system can make recommendations that are especially likely to appeal to this particular user.

Concepts

The key idea embodied in Amazon.com's recommendations and the other systems discussed in this chapter is that of adaptation to the individual user. Depending on their functions and forms, systems that adapt to their users have been given labels ranging from *adaptive interfaces* through *user modeling systems* to *software agents* or *intelligent agents*. Starting in the late 1990s, the broader term *personalization* became popular, especially in connection with commercially deployed systems. In order to discuss the common issues that all of these systems raise, we will refer to them with the term *user-adaptive systems*, which describes their common properties explicitly. Fig. 22.2 introduces some concepts that can be applied to any user-adaptive system; Fig. 22.3 shows the form that they take in Amazon.com's recommendations.

A user-adaptive system makes use of some type of information about the current individual user, such as the products that the user has bought. In the process of *user model acquisition*, the system performs some type of learning and inference on the basis of the information about the user in order to arrive at some sort of user model, which in general concerns only limited aspects of the user (such as his or her interest in particular types of product). In the process of *user model application*, the system applies the user model to the relevant features of the current situation in order to determine how to adapt its behavior to the user.

A user-adaptive system can be defined as an interactive system that adapts its behavior to individual users on the basis of processes of user model acquisition and application that involve some form of learning, inference, or decision making.

This definition distinguishes user-adaptive systems from *adaptable* systems: ones that the individual user can explicitly tailor to his or her own preferences (for example, by choosing options that determine the appearance of the user interface). The relationship between adaptivity and adaptability will be discussed at several places in this chapter.



FIGURE 22.1. Part of a screen showing a list of recommendations generated on request by Amazon.com. (Screen shot made from http://amazon.com/ in December 2005.)

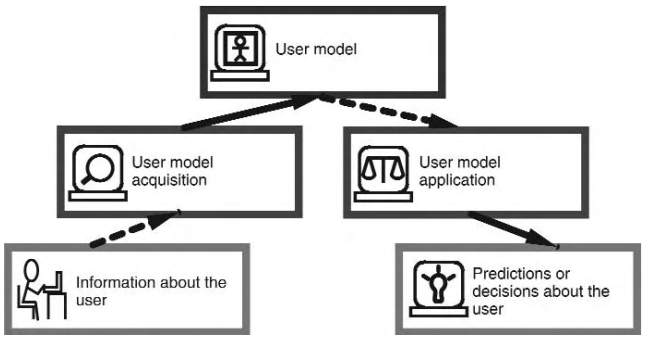


FIGURE 22.2. General schema for the processing in a user-adaptive system. (Dotted arrows: use of information; solid arrows: production of results.)

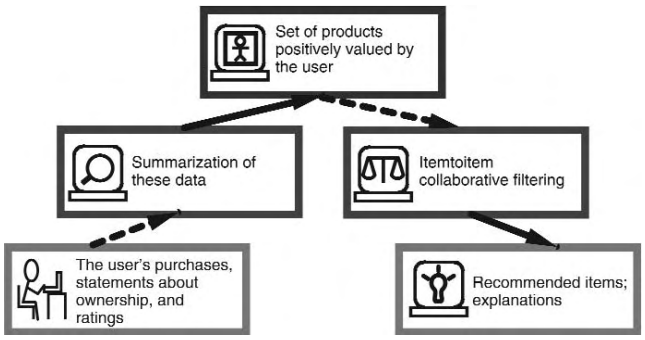


FIGURE 22.3. Overview of adaptation in Amazon.com.

The next two sections of this chapter address the following question: What can user-adaptivity be good for? They examine in turn 10 different functions that can be served by user-adaptivity, giving examples ranging from familiar commercially deployed systems to research prototypes. The following section discusses some usability challenges that are especially important in connection with user-adaptive systems and that have stimulated most of the controversy that has surrounded these systems. The next section considers a key design decision: What types of information about each user should be collected? The final major section looks at several distinctive aspects of the empirical study of user-adaptive systems. The chapter concludes with comments on the reasons why their importance is likely to continue to grow.[1]

FUNCTIONS: SUPPORTING SYSTEM USE

Some of the ways in which user-adaptivity can be helpful involve support for a user's efforts to operate a system successfully and effectively. This section considers five types of support.

Taking Over Parts of Routine Tasks

The first function of adaptation involves taking over some of the work that the user would normally have to perform himself or herself—routine tasks that may place heavy demands on a user's time, though typically not on his or her intelligence and knowledge. Maybe the most obvious task of this sort is organizing e-mail, which takes up a significant proportion of the time of many office workers. The classic early work of Maes's (1994) group on "agents that reduce work and information overload" addressed this task.

A more recent effort (Fig. 22.4) is found in the prototype "intelligent electronic mail sorter" (i-ems; McCreath, Kay, & Crawford, 2006; see also Crawford, Kay, & McCreath, 2002; McCreath & Kay, 2003), which is designed to expedite the tedious task of filing incoming e-mail messages into folders. By observing and analyzing the way an individual user files messages, the system learns to predict the most likely folder for any new message. In the overview of messages in the user's inbox (shown in the top part of the screen shot), i-ems tentatively sorts the new messages into categories that correspond to the most likely folders. When an individual message is being displayed, the one-line field in the middle of the screen shows an explanation of the folder prediction. If the user agrees with the prediction, he or she can click on the "Archive" button at the top of the screen to have the message moved into the predicted folder. To file it away in another folder, he or she drags it to the icon for the folder in the left-hand panel, just as he or she would with a system that did not make any predictions.

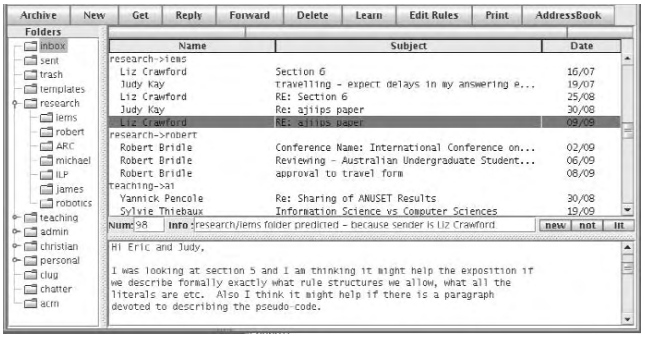


FIGURE 22.4. Partial screen shot from the intelligent e-mail sorting system i-ems. (Screen shot courtesy of Eric McCreath.)

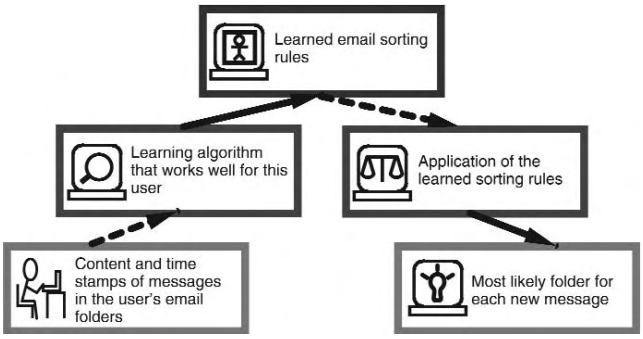


FIGURE 22.5. Overview of adaptation in i-ems.

One reason why research on systems like i-ems has continued for so long is that the problem raises a number of challenges. For example, since different users apply radically different principles for creating categories of e-mail, i-ems supplies several different methods for learning a user's implicit rules, each of which may show a different degree of success with different users. It also allows the learned rules to operate alongside any handcrafted rules that the user may have defined so that the strengths of both types of rules can be exploited (an evaluation is discussed below in the empirical methods section). Since even the best set of learned rules will sometimes incorrectly predict how the user would classify a message, the interface must be designed in such a way that incorrect predictions will have minimal consequences. New approaches to the general problem continue to appear (e.g., Surendran, Platt, & Renshaw, 2005). Two systems that have been fairly widely deployed have been SwiftFile (Segal & Kephart, 1999, 2000), which was incorporated into Lotus Notes, and POPFile (available in early 2006 from http://popfile.sourceforge.net/cgi-bin/wiki.pl), a public domain program that is used mainly for spam filtering but that can also learn to sort messages into a limited number of user-specific folders.[2]

[1]The version of this chapter in the first edition of this handbook included a section about some of the machine learning and artificial intelligence techniques that are most commonly used for user model acquisition and application. There is no such section in this second edition, because (a) it seemed more important to expand the material in the other sections and (b) the range of techniques used has grown to the point where a brief summary would have limited value. Discussions of the relevant methods will be found in many of the works cited in the chapter.

[2]For examples of approaches to support for e-mail management that do not involve adaptation to individual users, see Gruen et al. (2004) and Bälter and Sidner (2002).

Another traditional task in this category is the scheduling of meetings and appointments (Mitchell, Caruana, Freitag, McDermott, & Zabowski, 1994; Maes, 1994; Horvitz, 1999; Gervasio, Moffitt, Pollack, Taylor, & Uribe, 2005). By learning the user's preferences for particular meeting types, locations, and times of day, a system can tentatively perform part of the task of entering appointments in the user's calendar.

The primary benefits of this form of adaptation are saving time and effort for the user. The potential benefits are greatest where the system can perform the entire task without input from the user. In most cases, however, the user is kept in the loop (as with i-ems), because the system's ability to predict what the user would want done is limited (cf. the section on usability challenges below).

Adapting the Interface

A different way of helping a person to use a system more effectively is to adapt the user interface so that it fits better with the user's way of working with the system. Interface elements that have been adapted in this way include menus, icons, and the system's processing of signals from input devices such as keyboards.

An example familiar to most readers is provided by the Smart Menus feature that has been found in Microsoft operating systems since Windows 2000. Fig. 22.6 illustrates the basic mechanism: An infrequently used menu option is initially hidden from view; it appears in the main part of a menu only after the user has selected it for the first time. (It will be removed later if the user does not select it often enough.) The idea is that, in the long run, the menus should contain just the items that the user accesses frequently (at least recently), so that the user needs to spend less time searching within menus.

Some informative studies related to Smart Menus have been conducted by McGrenere, Baecker, and Booth (2002). In a field

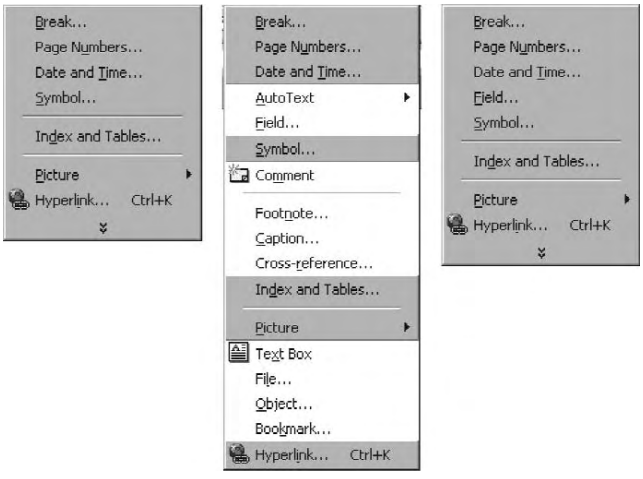


FIGURE 22.6. Example of adaptation in Smart Menus. (The user accesses the "Insert" menu. Not finding the desired option, the user clicks on the extension arrows and selects the "Field" option. When the user later accesses the same menu, "Field" now appears in the main section.)

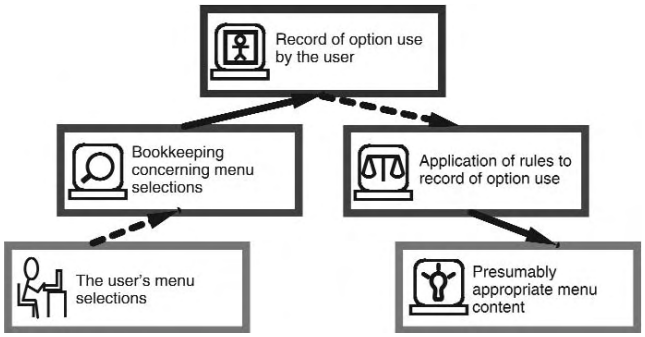


FIGURE 22.7. Overview of adaptation in Smart Menus.

study with experienced users of Microsoft Word 2000, McGrenere et al. compared the Smart Menus of Word 2000 with (a) traditional static menus and (b) an alternative approach to reducing the number of functions that confront users. Their variant MSWord Personal is an adaptable system: It provides a reasonably intuitive and convenient way for users to add and remove menu functions. After working with MSWord Personal for several weeks, most of the users in the study preferred this adaptable system to the normal Word 2000 with Smart Menus. The users who had been classified as feature-shy appeared to benefit most, but as is typical in studies like this (as will be discussed), quite a variety of attitudes about the relative merits of the three approaches to adapting menu content were shown by the subjects. As the authors pointed out, it seems worthwhile to consider design solutions that combine some degree of adaptivity and adaptability. For example, instead of automatically adapting the menus, the system might recommend possible adaptations on the basis of its analysis of the user's menu selections (cf. Bunt, Conati, & McGrenere, 2004).

A more direct experimental comparison by Findlater and McGrenere (2004) involving adaptive menus like Smart Menus is discussed in the section on empirical methods below.

One promising application of both adaptable and adaptive methods involves taking into account special perceptual or physical impairments of individual users to allow them to use a system more efficiently, with minimal errors and frustration. A system in which the two approaches are combined is the Web Adaptation Technology of IBM Research (Hanson & Crayne, 2005), which aims to facilitate web browsing by older adults. With regard to most of the adaptations, such as the reformatting of multicolumn text in a single column, the system is adaptable: It provides convenient ways for the user to request the changes. (It would in fact be difficult for a system to determine automatically whether a given user would benefit from one-column formatting.) Several changes in the keyboard settings are achieved via automatic adaptation (for a more detailed discussion, see Trewin, 2004). For example, the *key repeat delay* interval is a parameter that determines how long a key (e.g., the left-arrow key) has to be held down before the system starts repeating the associated action (e.g., moving the cursor to the left). Some users require a relatively long key repeat delay because of a tendency to hold keys down relatively long even when they do not want repetition. But asking the user to specify the key repeat delay is not an attractive option: It can be time-consuming to

explain what the parameter means; the user him- or herself may have no idea what the best setting is for him or her; trial and error with different settings can be time-consuming and frustrating; and for some users, the optimal setting can change from day to day. The Dynamic Keyboard component of the Web Adaptation Technology therefore includes an algorithm that analyzes a user's typing behavior to determine an optimal key repeat delay (as well as other parameters); the system then adjusts the parameter in a relatively conservative fashion. Although, under some circumstances, automatic adjustment of keyboard parameters could make the keyboard unpredictable and hard to use, results obtained in the context of Web Adaptation Technology (Trewin, 2004) revealed no problems of this sort.

Helping With System Use

Instead of suggesting (or executing) changes to the interface of a given application, a user-adaptive system can adaptively offer information and advice about how to use that application, and perhaps also perform some of the necessary actions itself. Various tendencies make it increasingly difficult for users to attain the desired degree of mastery of the applications they use. A good deal of research into the development of systems that can take the role of a knowledgeable helper was conducted in the 1980s, especially in connection with the complex operating system Unix.[3] During the 1990s, such work became less frequent, perhaps partly because of a recognition of the fundamental difficulties involved. In particular, it is often difficult to recognize a user's goal when the user is not performing actions that tend to lead toward that goal. The Office Assistant, an ambitious attempt at adaptive help introduced in Microsoft Office 97, was given a mixed reception, partly because of the inherent difficulty of its task but especially because of its widely perceived obtrusiveness (cf. the section on usability challenges below).

Most adaptive help systems to date have been based on the paradigm called *keyhole recognition*: (passively) observing the user and attempting to make useful inferences about her goals and tasks. By contrast, Fig. 22.8 shows an example of an alternative approach to intelligent help that has been developed by researchers at Mitsubishi Electric Research Laboratory, which is based on a collaborative dialogue paradigm (Rich et al., 2005; for a presentation of the theoretical and technical background, see Rich & Sidner, 1998; Rich, Sidner, & Lesh, 2001). In this demonstration scenario, DiamondHelp is collaborating with the user of a feature-rich programmable washer-dryer.[4] Instead of working independently on the problem, the user conducts a dialogue with the help system, the goal of the dialogue being the execution of the user's task. The dialogue contributions of the help system and the user are shown in the chat balloons at the left- and right-hand sides, respectively, of the screen. The user's possible dialogue contributions are automatically generated from the current collaborative dialogue state and offered in a menu inside of his or her balloon. The user can choose what he or she wants to say by either touching the appropriate phrase or saying it using speech recognition.[5] For example, in the top part of the figure, the user is offered a choice among three possible top-level tasks, the most complex of which is defining a new cycle. In a typical exchange, the user specifies a goal or subgoal that he or she would like to achieve, and the system responds by giving instructions and perhaps offering further possible utterances for the user.

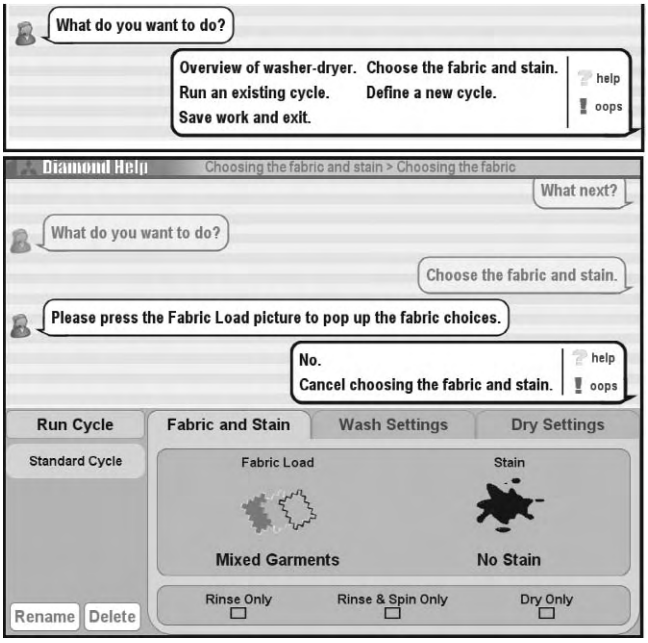


FIGURE 22.8. Example of collaborative assistance offered by DiamondHelp. (Explanation in text. Screen shots courtesy of Charles Rich.)

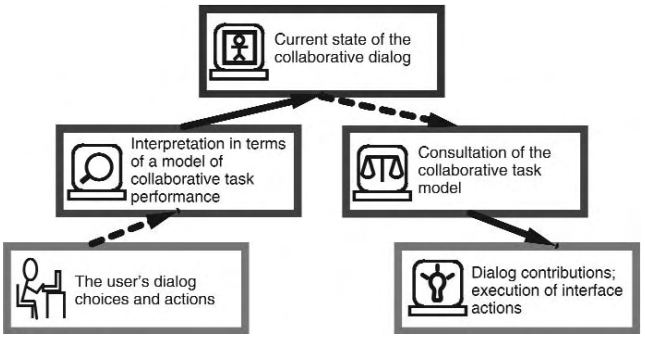


FIGURE 22.9. Overview of adaptation in DiamondHelp.

[3]A collection of papers from this period appeared in a volume edited by Hegner, McKevitt, Norvig, and Wilensky (2001).

[4]The interface shown in the figures may be displayed on the washer-dryer itself or remotely accessed via a home network.

[5]DiamondHelp does not support unrestricted natural language or speech understanding. In a Wizard of Oz study (a type of study that will be discussed in the section on empirical methods) involving a prototype help system of this general sort, DeKoven (2004) found that users who were able to employ unrestricted speech would have preferred to have more guidance about what they could say to the system.

This dialogue in some ways resembles the interaction with the more familiar type of wizard that is often employed for complex, unfamiliar tasks such as software installation. The main difference is that the dialogues with DiamondHelp can be more flexible, because the system has explicit models of the tasks that the user can perform and is capable of making use of these models in various ways during the dialogue. For example, after pressing the Fabric Load picture, the user can continue manipulating the GUI in the lower half of the screen by himself or herself until he or she requests guidance again (e.g., by asking "What next?"). Because the user's actions with the interface are reported to the help system, the help system can keep track of how far the user has progressed in the performance of his or her task. In other words, the help system incorporates a restricted form of the sort of goal and plan recognition that featured prominently in earlier intelligent help systems. In DiamondHelp, recognition of the user's actions is relatively likely to be accurate, because of the information that the user has supplied about his or her goals (e.g., Lesh, Rich, & Sidner, 1999). Depending on the experience and the preferences of the user, therefore, the user can rely on the help system to various degrees, ranging from ignoring it, occasionally asking for a hint, or being led step by step through the entire task.

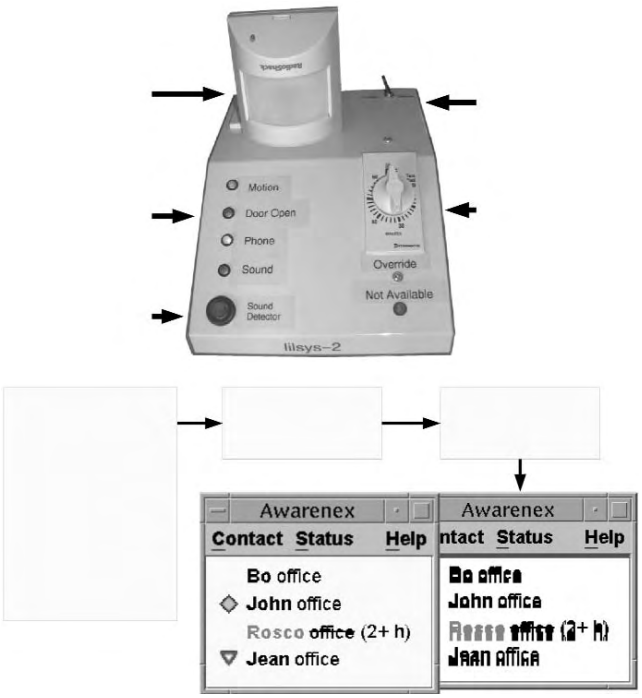


FIGURE 22.10. Above: Lilsys's sensor and data acquisition module; below: Lilsys's data flow and a screen shot of the user interface. (Adapted from Figures 1 and 2 of: "Lilsys: Sensing unavailability," by J. Begole, N. E., Matsakis, & J. C. Tang, 2004, In J. Herbsleb & G. Olson (Eds.), *Proceedings of the* 2004 *Conference on Computer-Supported Cooperative Work,* pp. 511–514, New York: ACM. Copyright 2004 by ACM. Adapted with permission.)

Mediating Interaction With the Real World

Whereas an intelligent help system aids the user as he or she uses a complex interactive system, some recently developed systems help the user to cope with the world itself. They do so by acquiring and processing evidence concerning the user's cognitive and emotional state and taking actions designed to mitigate any conflict between this state and the demands of the environment.

One common function of systems in this category is to protect people from the flood of incoming messages (via cell phone, instant messaging, e-mail, and other channels) whose number and diversity are increasing with advances in communication technology. When a potential recipient is focusing on some particular task or activity, an adaptive assistant causes messages to be discouraged, delayed, or otherwise buffered until some more appropriate time. One strategy is to provide to the potential initiators of communication information about the state of the recipient. The experimental prototype Lilsys (cf. Fig. 22.10) illustrates this strategy. The system continuously updates a user model that contains assessments of its user's availability for communication. The assessments are based on a number of cues that have been found in previous research to be useful predictors of a person's physical presence and availability: whether the user (or someone else in the room) is moving or speaking; whether the door is open; whether the user is using the phone or the computer keyboard and mouse; and what events are scheduled in the user's calendar. A handcrafted model uses this information to arrive at a global assessment of the user's availability; this assessment is in turn displayed to potential communicators.[6] A field study with a small number of users indicated that other persons do in fact adapt their behaviors to take into account a Lilsys user's availability, often by changing the nature of their communications rather than by postponing it. Lilsys users appreciated the possibility of having their availabili-

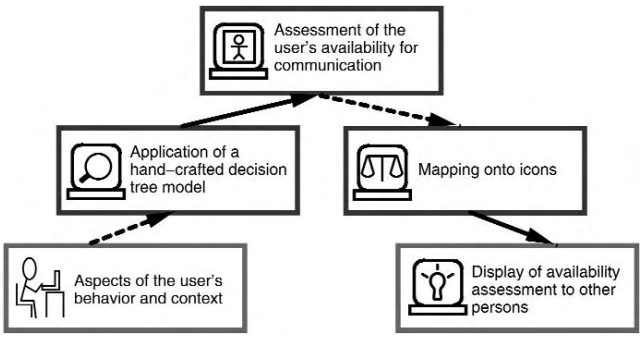


FIGURE 22.11. Overview of adaptation in Lilsys.

[6]In many systems in this category, such as the ones mentioned in the following paragraph, the model for the interpretation of evidence is acquired via machine learning methods on the basis of relevant training data.

ties sensed automatically, as opposed to having explicitly to specify availability themselves. For example, they virtually never used the timer switch (visible in Fig. 22.10) that allowed them to specify that they were going to be unavailable for a particular period. More generally, the modeling of a user's changing cognitive or emotional state appears to be a task for which automatic adaptation is an especially promising approach, simply because people are typically not willing or able to update an explicit self-assessment continually.

In some other availability management systems, decisions about when and how to present messages are made by the system itself on the basis of the user model (e.g., Horvitz, Koch, Sarin, Apacible, & Subramani, 2005). A good deal of research has examined effective cues for the recognition of availability and interruptibility (e.g., Fogarty et al., 2005; Ho & Intille, 2005; Iqbal, Adamczyk, Zheng, & Bailey, 2005).

A related line of research has focused on the recognition of the mental states of drivers, which is especially important because of safety issues. The modeling of interruptibility is important here as well (e.g., Schneider & Kiesler, 2005). But even when no other persons are involved, there are reasons to try to recognize safety-relevant states, like drowsiness and stress, so that the system can intervene, for example, by waking the driver up or by playing soothing music. Stress and emotion are manifested in physiological indicators (e.g., Healey & Picard, 2000; Lisetti & Nasoz, 2004) and in speech (e.g., Fernandez & Picard, 2000; Jones & Jonsson, 2005). Products along these lines have begun to appear in cars, beginning with relatively simple detection methods such as the recognition of long or frequent eye closures. An example of a more complex and comprehensive approach to the modeling of drivers' affective state is found in the work of Li and Ji (2005).

A general problem with adaptation for the purpose of safety is that the user may come to rely on the adaptation, reducing his or her own attention to safety.[7] For example, a driver may make less effort to avoid distraction or to remain alert, expecting that the assistant will recognize any dangerous situation and warn him or her in time. Especially since the recognition of a person's mental states is almost always error-prone, this tendency can eliminate the potential safety benefits of monitoring systems unless appropriate measures are taken (e.g., making warning sounds so unpleasant that the driver will want to avoid relying on them more than is necessary).

Controlling a Dialogue

Much of the early research on user-adaptive systems concerned systems that conducted natural language dialogues with their users (see, e.g., Kobsa & Wahlster, 1989). During the 1990s, attention shifted to interaction modalities that were more widely available and that made it possible in many cases to implement adaptation straightforwardly. Toward the year 2000, advances in the technology of natural language and speech processing led to a recent reawakening of interest in user-adaptive dialogue systems (see, e.g., Haller, McRoy, & Kobsa, 1999; Zukerman & Litman, 2001; Litman & Pan, 2002).

Natural language dialogue has served as an interaction modality in connection with most of the functions of user-adaptivity discussed in this and the following sections, such as the provision of help and the recommendation of products. One type of adaptivity is largely characteristic of natural language dialogue: adaptation of the system's dialogue strategy, which is a policy for determining when and how the system should provide information, acquire information from the user, and perform other dialogue acts.

Adaptation is especially important in spoken dialogue systems, such as those that offer information about train departures or flight arrivals via the phone. Novice users may require extensive explanations and frequent confirmations, but these elements can be unnecessarily time consuming and frustrating to experienced users. Many deployed systems apply simple adaptation principles that distinguish between new and more experienced users. For example, if a phone-based mail-order system knows that the current caller has previously ordered a product, it may adopt a dialogue style that presupposes familiarity with the system. Since it is not always this easy to classify the user, and since asking the user for a self-assessment can be awkward and time consuming, researchers have looked for ways of adapting to the user on the basis of his or her behavior during the current dialogue (e.g., Litman & Pan, 2002). A recent example is shown in Fig. 22.12, which shows a translation of an example dialogue conducted with the Kyoto City Bus Information System (Komatani, Ueno, Kawahara, & Okuno, 2005). The system adjusts its assessments of three properties of the caller on the basis of each utterance of the caller: level of skill at conducting dialogues with this system; level of knowledge concerning the domain (e.g., Kyoto and its buses); and desire to complete the dialogue quickly ("urgency"). The evidence used is a set of properties such as the amount of information specified in each utterance; whether the user refers to bus stops by their exact names; and whether the user barges in, interrupting the system before the system has completed an utterance. The dialogue excerpt includes several examples that illustrate how the system's dialogue behavior is adapted on the basis of the user model. In a controlled experiment, each of 20 subjects conducted dialogues with the system either with the adaptive version or with a version that assumed that all callers had the same properties. Dialogues were completed more quickly with the adaptive variant, and subjects expressed greater satisfaction.[8]

Another popular idea about how to adapt dialogues to the user concerns the recognition of negative emotions like anger and frustration, the goal being to transfer callers who express these emotions to human call agents before they are lost as customers. Although there has been a lot of research on the recognition of mental states on the basis of speech in dialogues

[7]A brief overview of the theory of *risk homeostasis* is given by Wilde (1998), while a recent alternative perspective is offered by Fuller (2005).

[8]See the section on empirical methods for a discussion of the difficulties involved in the interpretation of the results of experiments like this one. Although the Kyoto City Bus Information System is accessible to the public, the adaptive features described here are normally turned off, because they lead to slower processing given the currently available infrastructure.

case:	[skill level: low] [knowledge level: low] [urgency: low]
Sys:	What is your current bus stop, your destination, or specific bus route number?
User:	Maruyama Park.
Sys:	Let me confirm these in turn.
	(← Explanation is added by [skill level: low].)
	Will you get on at Maruyama Park?
	(← Content is confirmed by [urgency: low].)
	Reply with yes or no.
	(← Instruction is added by [skill level: low].)
User:	Yes.
Sys:	Where will you get off the bus?
	Specify by bus stop name or a well-known landmark.
	(← Explanation is added by [skill level: low].)
User:	Tofukuji Temple.
Sys:	Will you get off the bus at Tofukuji Temple?
	Reply with yes or no.
User:	Yes.
Sys:	Following buses are located.
	(← Explanation is added by [skill level: low].)
	Nearest bus stop to Maruyama Park is Gion.
	(← Domain information is added by [knowledge level: low].)
	Bus number 202 bound for Nishioji-Kujo has departed Higashiyama-Sanjo, which is two stops away.
	. . .

FIGURE 22.12. Excerpt from a dialog with the Kyoto City Bus Information System. (Figure 2 of "User modeling in spoken dialogue systems to generate flexible guidance," by K. Komatani, S. Ueno, T. Kawahara, & H. G. Okuno, 2005, *User Modeling and User-Adapted Interaction,* 15, pp. 169–183. Copyright 2005 by Springer Verlag. Reproduced with permission.)

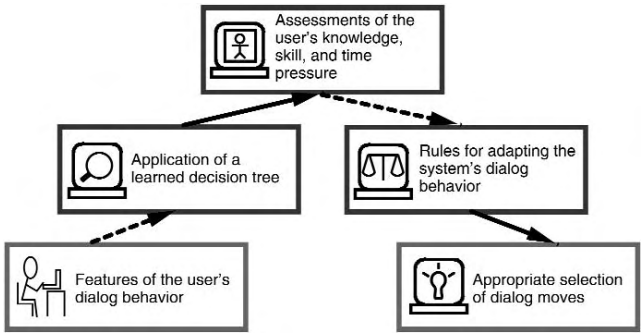


FIGURE 22.13. Overview of adaptation in the Kyoto City Bus Information System.

(e.g., Yacoub, Simske, Lin, & Burns, 2003; Liscombe, Riccardi, & Hakkani-Tür, 2005), it remains to be seen how widespread this particular application will become. One possible drawback is that with some systems callers might find it worthwhile to adapt to the adaptation (as with safety-relevant adaptations), feigning emotion in order to get quicker attention.

Other dialogue adaptations that are being explored concern stable personal characteristics like gender and age. Since it is possible to recognize these characteristics reasonably well on the basis of speech, a system might adopt a voice or dialogue style that designers thought appropriate for the age and gender in question (e.g., Müller, Wittig, & Baus, 2003).

FUNCTIONS: SUPPORTING INFORMATION ACQUISITION

We are constantly hearing that information overload is a typical problem of our age, especially because of the explosive growth of the Internet and in particular the World Wide Web. In addition to the vast number of electronic documents of various sorts, users now have access to a vast number of products available for sale, people that they can get in touch with, and systems that can teach them about some topic. The second major type of function of user-adaptive systems is to help people to find what they need in a form that they can deal with.

Helping Users to Find Information

We will first look at the broad class of systems that help the user to find relevant electronic documents, which may range from brief news stories to complex multimedia objects.

As an especially clear example, consider the situation of a user who, in the year 2006, heard that a lot of interesting facts and opinions can be found in weblogs (blogs), of which millions are accessible. The user would like to be able to read, each day, the articles in blogs that are of particular interest. But how is he or she to find these? The user does not know in advance which blogs are especially likely to produce material of interest (as, for example, he or she could specify a well-known newspaper as a promising source of online stories if he or she were interested in news). The user could submit queries to a search engine that indexes blogs, but in general, he or she cannot know in advance what topics of interest will be covered by the latest blog articles; and given the low quality and lack of authority of many blogs, the user will not be confident of receiving good results on any given topic.

An approach to this problem that relies heavily on adaptation to the individual user (called "personalization" in this context) is realized at the time of this writing in the Website Findory (http://findory.com), which offers access to both blogs and news articles. To the new user visiting the blog section of the website, Findory offers a page that shows the first few lines of a number of blog articles on different topics (cf. Fig. 22.14). The user can then click to read the articles that interest him or her most. Each selection causes the system to update its model of the user's interests and adapt the selection of blog articles accordingly. For example, if the user chooses an article discussing a copyright infringement suit against a search engine company, further articles concerning copyright issues and search engines are likely to appear, marked with the sunburst icon that is visible in Fig. 22.14. If the user clicks on the icon for a recommended article, a page is displayed that explains the recommendation in a style similar to that of Amazon.com (cf. Fig. 22.1 and the next subsection) with a list of articles that the user has read in the past which are similar to the recommended item in terms of their content (which the system can characterize on

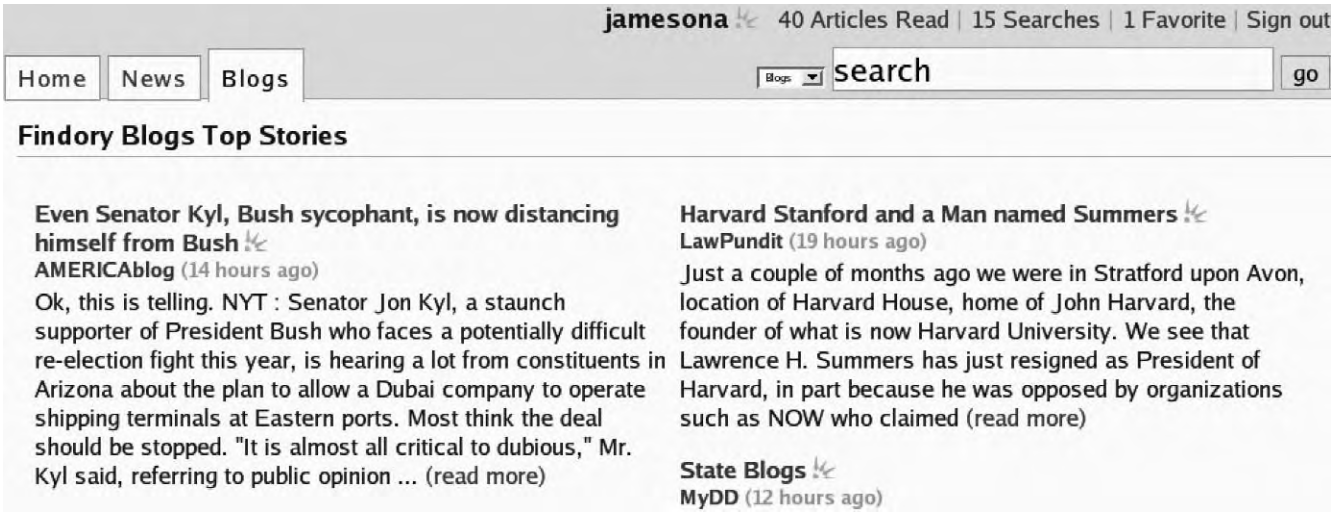


FIGURE 22.14. A small part of a personalized display of Findory (http://findory.com, March 2006). (The icon that appears after a title indicates that the entry has been recommended on the basis of the articles that the user has read previously.)

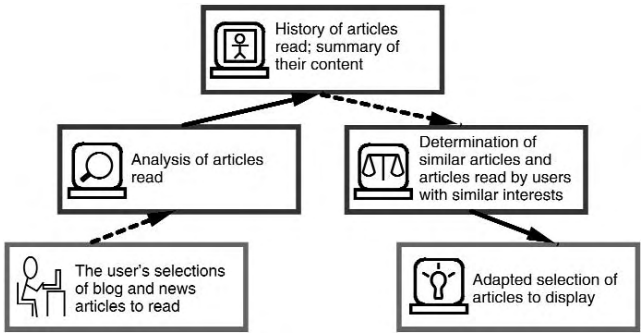


FIGURE 22.15. Overview of adaptation in Findory.

the basis of the words in the text) or in terms of the users who have previously read them (G. Linden, personal communication, February, 2006). If the user sees in this way an article that he or she does not want to be used for recommendations in the future, he or she can delete it from the reading history.

Findory's approach relies on the user being able to identify, early in his or her use of the system, some articles that interest him or her and that can therefore serve as examples for the system's learning. This process is facilitated by the opportunities that the user has to issue explicit queries with keywords and to consult trusted sources in the News section of the website. The main advantage of this approach is that the user need not make any effort to specify explicitly in what types of content he or she is interested. Any such effort would be problematic anyway in that (a) it can be difficult and tedious to describe a large number of general interests accurately, for example by specifying relevant key words; and (b) since interests change over time and as a function of current developments, the user would have to keep updating the descriptions.

More generally speaking, user-adaptive systems that help users find information[9] typically draw from the vast repertoire of techniques for analyzing textual information (and to a lesser extent, information presented in other media) that have been developed in the field of information retrieval. The forms of adaptive support are in part different in three different situations, the first two of which can arise with Findory:

Support for browsing. In the World Wide Web and other hypermedia systems, users often actively search for desired information by examining information items and pursuing cross-references among them. A user-adaptive hypermedia system can help focus the user's browsing activity by recommending or selecting promising items or directions of search on the basis of what the system has been able to infer about the user's information needs. An especially attractive application scenario is that of mobile information access, where browsing through irrelevant pages can be especially time-consuming and expensive. In this context, the best approach may be for the system to omit entirely links that it expects to be less interesting to the individual user. Billsus and Pazzani (2007) describe a case study of an adaptive news server that operates in this way. Stationary systems with greater communication bandwidth tend to include all of the same links that would be presented by a nonadaptive system, highlighting the ones that are most likely to be of interest or presenting separate lists of recommended links. As is argued and illustrated by Tsandilas and schraefel (2004), this approach makes it easier for the user to remedy incorrect assessments of the user's interests on the part of the system.

[9]Surveys of parts of this large area are provided by, among others, Kelly and Teevan (2003) and several chapters in the collection edited by Brusilovsky, Kobsa, and Nejdl (2007).

Support for query-based search or filtering. Web search engines have been enormously successful and popular in this context, but they have usually exhibited one limitation: The results presented for a given query have not depended on the interests or previous behavior of the individual user. By contrast, with *personalized search*, the search engine keeps track of user's search history, builds up some sort of model of the user's interests (either by keeping track of and analyzing the user's search history or by asking for an explicit description of interests), and biases the results presented accordingly by reordering or filtering the results. A good deal of research (e.g., Teevan, Dumais, & Horvitz, 2005; Micarelli, Gasparetti, Sciarrone, & Gauch, 2007) has demonstrated the potential benefits of this strategy. During the year before the writing of this chapter, a personalized variant of the search engine Google was introduced that sometimes reranked search results on the basis its record of the user's previous web searching behavior. However, it is unclear how widespread this approach will become. The added value of personalization is less obvious when the user has given an explicit query than when the user is simply looking for something interesting, as is often the case with Findory. With an explicit query, it is feasible and worthwhile for the user to think about an informative description of his or her interests and to modify his or her query (perhaps repeatedly) on the basis of the results obtained.

An interesting variant on personalized search is found in the system I-Spy (e.g., Smyth et al., 2005): This community-oriented search engine tailors the results of web search queries to an entire community of users, such as the employees of a particular company. It moves upward in the search results list results that have been clicked on by previous users in the community who had issued the same or similar queries.

Spontaneous Provision of Information

A number of systems present information that may be useful to the user even while the user is simply working on some task and is making no effort to find information. A recent prototype that has been deployed at a research laboratory is the FXPAL Bar (Billsus, Hilbert, & Maynes-Aminzade, 2005). While an employee visits websites in the course of normal work, the system searches in the background for potentially relevant information (e.g., about company visitors and internal publications). A central design issue for this and similar systems, concerns the methods for making the retrieved information available to the user. Presentation of results via means like pop-up windows risks being obtrusive (cf. the section on usability challenges below), but if the presentation is too subtle, users will often ignore the recommendations and derive little or no benefit from the system. Moreover, the optimal solution in general differs from one user to the next. Billsus et al. (2005) reported studies with a variety of interface solutions for the FXPAL Bar, some of which are adaptable by the user (e.g., the size of a translucent pop-up window that describes a potentially relevant document).[10]

Recommending Products

One of the most practically important categories of user-adaptive systems today comprises the product recommenders that are found in many commercial websites. The best-known system, the recommender of Amazon.com, was discussed briefly in the introduction to this chapter. Looking more closely at Fig. 22.1, we can see some distinguishing features of this approach to recommendation. As can be seen from the brief explanations that accompany the recommendations, the system takes as a starting point the information it has about the user's ownership or evaluation of particular products. It then recommends products that are similar in the sense that there is large overlap in the sets of customers that buy them (hence the familiar explanations of the form "Customers who bought this title also bought . . ."). That is, the recommendations are based on a statistical analysis of purchases made by many users, an approach known as *collaborative filtering* (see, e.g., Schafer, Frankowski, Herlocker, & Sen, 2007, for an overview). The products recommended in this way may also be similar in the sense of having the same author or a similar title (as in the examples in the figure), but similarities of this sort can also be conspicuously absent: In the category "Coming soon," the user of Fig. 22.1 was recommended the DVD *The Island* because of having positively rated a Sony VAIO notebook PC. As is explained by Linden, Smith, and York (2003), the details of this particular variant of collaborative filtering are due largely to the constraint that it has to be able to cope with Amazon.com's millions of products and customers. Although it is generally acknowledged that the recommendations are not always accurate, they can yield notable benefits simply by being better than the generic recommendations (e.g., of top-selling items) that would be presented without personalization.

Some product recommenders allow and require the user to specify his or her evaluation criteria explicitly, instead of simply rating or purchasing individual items. For example, with the Active Buyers Guide in Fig. 22.16, the user has specified how he or she intends to use the digital camera that he or she would like to buy, and the system has recommended three cameras, explaining why each one is suitable in terms of the user's requirements. This needs-based approach to recommendation offers a natural alternative to the purely statistical approach of systems like Amazon.com when relatively complex and important decisions are involved for which it is worthwhile for the user to think carefully about the attributes of the products in question. However, it does require that a good deal of knowledge about the features of products and their relationships to user requirements be incorporated in the system.

An intermediate approach between these two extremes is the critiquing paradigm (for an early exposition, see Burke, Hammond, & Young, 1997; for an evaluation of some recent advances, see McCarthy, Reilly, McGinty, & Smyth, 2005). The distinguishing feature is an iterative cycle in which the system proposes a product (e.g., a restaurant in a given city), the user criticizes the proposal (e.g., asking for a more casual restaurant), and the system proceeds to propose a similar product that takes the critique into account.

[10]Influential earlier systems in this category include those of Rhodes (2000) and Budzik, Hammond, and Birnbaum (2001).

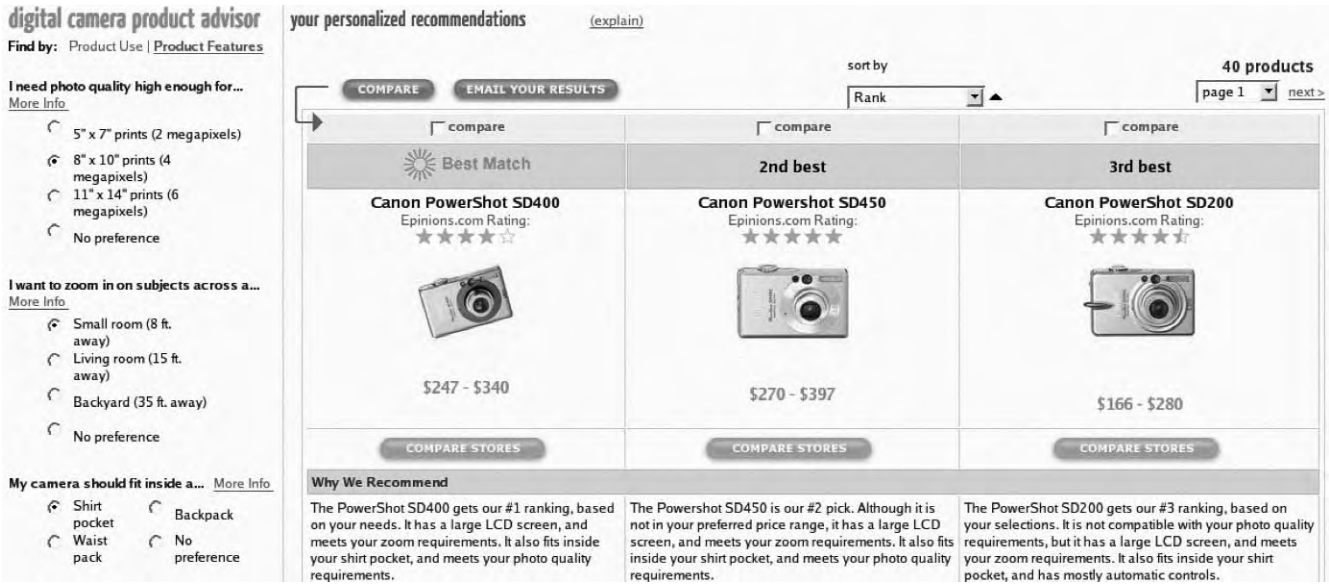


FIGURE 22.16. Partial screen shot from the Active Buyers Guide recommender for digital cameras. (Screen shot made from http://www.activebuyersguide.com in December 2005.)

Since some products are often used by groups of users (e.g., movies, vacations), a number of systems have been developed that explicitly address groups (for an overview, see Jameson & Smyth, 2007). The need to address a group rather than an individual has an impact on several aspects of the recommendation process. Users may want to specify their preferences in a collaborative way; there must be some appropriate and fair way of combining the information about the various users' preferences; and the explanations of the recommendations may have to refer to the preferences of the several individual group members.

Product recommenders of these various types address several problems that computer users typically experience when they search for products:

1. The user may not know what aspects of the products to attend to or what criteria should determine his or her decision. Some recommenders either (a) make it less necessary for the user to be explicitly aware of his or her evaluation criteria (as when collaborative filtering is used) or (b) help the user to learn about his or her own criteria during the course of the interaction with the system.
2. If the user is unfamiliar with the concepts used to characterize the products, he or she may be unable to make effective use of any search or selection mechanisms that may be provided. Product recommenders generally reduce this communication gap by allowing the user to specify his or her criteria (if this is necessary at all) in terms that are more natural to him or her. For example, in Fig. 22.16, the user does not need to know in advance how many megapixels his or her camera requires, since the question about photo quality is formulated in everyday terms.
3. The user may have to read numerous product descriptions in various parts of the website, integrating the information found in order to arrive at a decision. Once a product recommender has acquired an adequate user model, the system can take over a large part of this work, often examining the internal descriptions of a much larger number of products than the user could deal with.

From the point of view of the vendors of the products concerned, the most obvious potential benefit is that users will find one or more products that they consider worth buying, instead of joining the notoriously large percentage of browsers who never become buyers. A related benefit is the prospect of cross-selling: the system's model of the user can be employed for the recommendation of further products that the user might not have considered him- or herself. Finally, some vendors aim to build up customer loyalty with recommenders that acquire long-term models of individual customers: If the user believes that the system has acquired an adequate model of him or her, the user will tend to prefer to use the system again rather than starting from scratch with some other system.

Tailoring Information Presentation

Even after a system has decided which documents or products to present to a user, the question may remain of exactly how to present them. In some cases, this question should be answered differently for different users. In Fig. 22.16, the verbal descriptions of the recommended products refer explicitly to the preferences that the user has expressed; if they did not, the user would have to invest more effort to judge how well each product met his or her requirements.

An example from a research prototype is found in the system RIA (Fig. 22.17; Zhou & Aggarwal, 2004), a multimodal system that helps users search for real estate, often presenting infor-

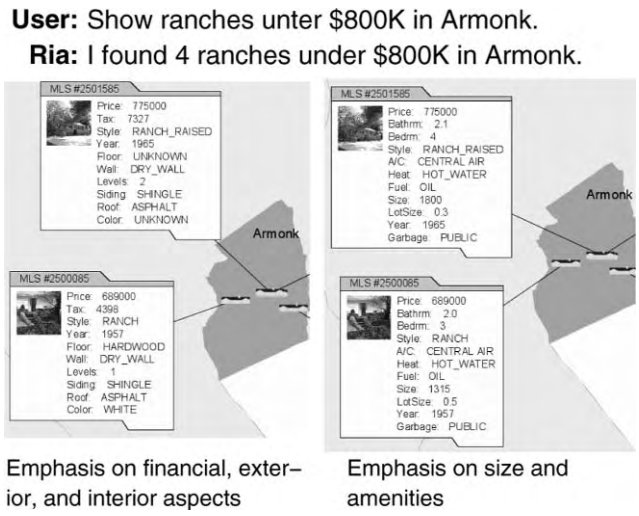


FIGURE 22.17. Two cropped screen shots from the RIA multimedia conversation system. (Screen shots courtesy of Vikram Aggarwal.)

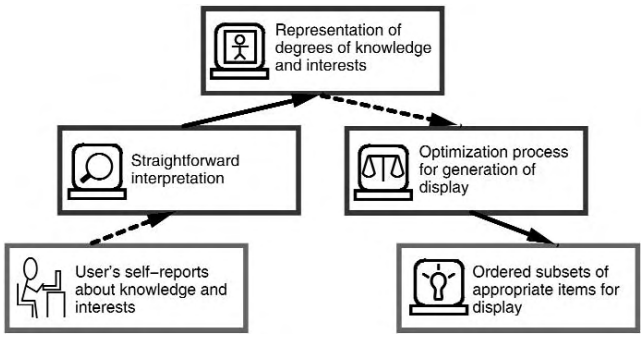


FIGURE 22.18. Overview of adaptation in RIA.

mation about houses on a map. As Fig. 22.17 shows, the amount of space available for describing a house is limited, so it is important to select the information that is most likely to help the user to decide how to proceed; otherwise, the user may have to request additional information explicitly, which would slow down the interaction. The process of user model acquisition (the three left-most nodes in Fig. 22.18) is quite straightforward in RIA: Before the interaction begins, the user is asked a small number of questions about his or her interest and knowledge concerning houses. The sophisticated part of the system is the way in which it uses this information—along with information about the houses that satisfy the query—to decide which information to select for presentation and how it is to be ordered. The problem is one of optimizing the presentation with respect to a large set of constraints.[11] In an evaluation, the displays generated by this method were similar to those generated by a human designers, who found the task of selecting the appropriate information items to be quite time-consuming (cf. the section on empirical methods for comments on this evaluation method).

Another class of systems in which the tailoring of information to individual users has promise comprises systems that present medical information to patients (for an overview, see Cawsey, Grasso, & Paris, 2007).

Properties of users that may be taken into account in the tailoring of documents include the user's degree of interest in particular topics; the user's knowledge about particular concepts or topics; the user's preference or need for particular forms of information presentation; and the display capabilities of the user's computing device (e.g., web browser vs. cell phone). One strong point of the optimization approach taken with RIA is that all of these factors can be represented and taken into account within a uniform framework.

Even in cases where it is straightforward to determine the relevant properties of the user, the automatic creation of adapted presentations can require sophisticated techniques of natural language generation (e.g., Bontcheva & Wilks, 2005) and multimedia presentation generation. Various less complex ways of adapting hypermedia documents to individual users have also been developed (see Bunt, Carenini, & Conati, 2007).

Supporting Collaboration

The increasing tendency for computer users to be linked via networks has made it increasingly feasible for users to collaborate, even in a spontaneous way and without prior acquaintance. A system that has models of a large number of users can facilitate such collaboration by taking into account the ways in which users match or complement each other.

A striking, though not typical, example is the system AgentSalon (Sumi & Mase, 2001, 2002), shown in Fig. 22.19. The system is used at conferences, in conjunction with a handheld guide (PalmGuide) that collects information about exhibits that the user has visited and ratings that he or she has given of them (the purpose within PalmGuide being to make recommendations to the user about other exhibits). When two visitors agree to work with AgentSalon, the information about them is transferred from their handhelds to AgentSalon. Like a host at a party, AgentSalon then looks for topics about which the two visitors might hold an interesting conversation, such as an exhibit about which they gave different ratings. The system tries to get a conversation going by having two animated agents simulate a conversation between these two visitors.

User modeling has been applied in connection with several (partially overlapping) types of collaboration:

- In computer-supported learning environments, in which the idea of *collaborative learning* has gained popularity in recent years (e.g., Soller, 2007)

[11]Decisions about what modalities to use for presentation—for example, text or speech output—are made in a similar way (cf. Zhou, Wen, & Aggarwal, 2005).

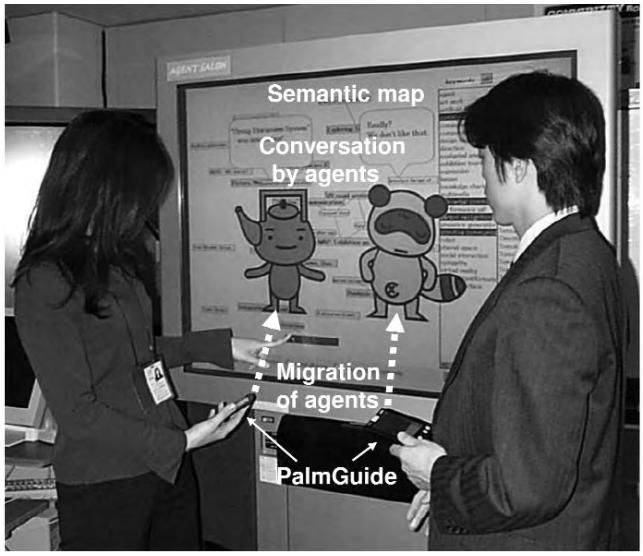


FIGURE 22.19. Attempt by AgentSalon to stimulate discussion between two conference visitors. (The system has identified an interesting topic by comparing the records of their conference experiences that have been stored on their PDAs. Figure 1 of "AgentSalon: Facilitating face-to-face knowledge exchange through conversations among personal agents," by Y. Sumi & K. Mase, 2001, in *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 393–400, New York: ACM. Research conducted at ATR Media Information Science Laboratories, Kyoto. Copyright 2001 by the Association for Computing Machinery, Inc. Reproduced with permission.)

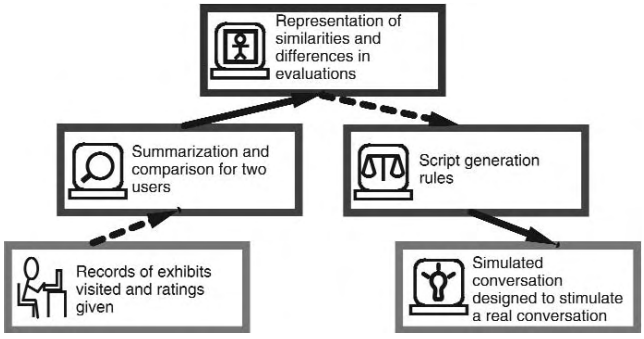


FIGURE 22.20. Overview of adaptation in AgentSalon.

- As a way of providing intelligent help for complex tasks (e.g., Vivacqua & Lieberman, 2000; Aberg & Shahmehri, 2001); putting a human expert into the loop is a way of avoiding some of the difficulties associated with fully automatic adaptive help systems that were discussed above
- In environments for computer-supported cooperative work within organizations (e.g., Terveen & McDonald, 2005)

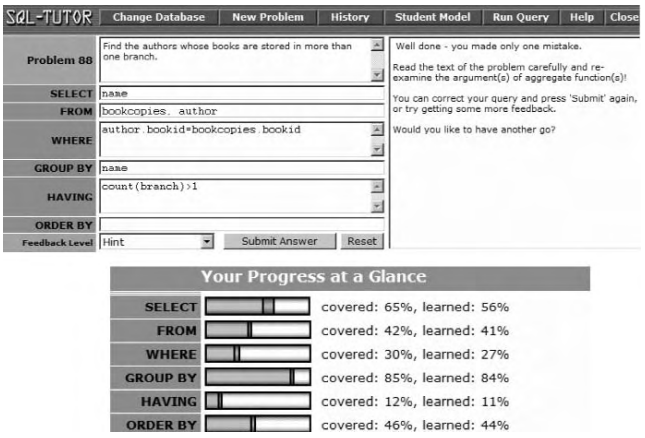


FIGURE 22.21. Screen shots from the SQL-Tutor. (Above: the main interface; below: display of the learner model. Screen shots courtesy of Antonija Mitrovic.)

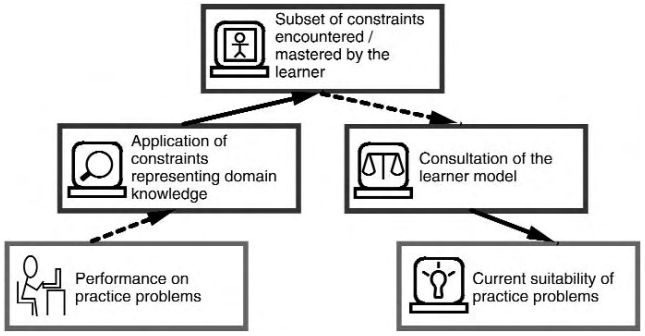


FIGURE 22.22. Overview of adaptation in the SQL Tutor.

Supporting Learning

Research on *student modeling*—or *learner modeling*, as it has been called more often in recent years—aims to add user-adaptivity to computer-based tutoring systems and learning environments (cf. Corbett, A. T., Koedinger, & Anderson, 1997).[12]

Increasingly, learning environments are being made available on the World Wide Web. An example is the SQL-Tutor (e.g., Mitrovic, Suraweera, Martin, & Weerasinghe, 2004), which teaches the database query language SQL.[13] The top part of Fig. 22.21 illustrates how the tutor presents a database querying problem and gives feedback on the learner's solution. The lower part of the figure shows a simple visualization of the learner model, which indicates the learner's degree of mastery of each of the six clauses of the SQL SELECT statement. It also shows a suggestion by the system about what type of problem the learner should attempt next—one of several ways in which the system helps the learner to pick a problem to work on next.

[12]Good sources of literature include the *International Journal of Artificial Intelligence in Education* and the proceedings of the biennial Conferences on Artificial Intelligence in Education (e.g., Looi, McCalla, Bredeweg, & Breuker, 2005).

[13]The tutor is available to registered students via Addison-Wesley's Website Database Place (http://www.aw-bc.com/databaseplace/).

A number of different aspects of the SQL-Tutor were evaluated in 10 studies (e.g., Mitrovic et al., 2004; Mitrovic & Ohlsson, 1999), including 2 studies that showed the value of helping learners to choose the next problem and showing learners their learner model.

Interaction in intelligent tutoring systems and intelligent learning environments can take many forms, ranging from tightly system-controlled tutoring to largely free exploration by the learner. Aspects of the system that can be adapted to the individual user include (a) the selection and the form of the instructional information presented, (b) the content of problems and tests, and (c) the content and timing of hints and feedback.

Learner-modeling systems may adapt their behaviors to any of a broad variety of aspects of the user, such as (a) the user's knowledge of the domain of instruction, including knowledge acquired prior to and during the use of the system, (b) the user's learning style, motivation, and general way of looking at the domain in question, and (c) the details of the user's current processing of a problem.

The underlying assumption is that the adaptation of the system's behavior to some of these properties of the learner can lead to more effective and more enjoyable learning. One series of studies that directly demonstrated the added value of learner-adaptive tutoring is described by A. Corbett (2001). Many evaluations, however, do not focus on measuring the benefits of adaptivity but rather on comparing alternative variants of the same adaptive system. In some cases, it has been found that the modeling of the learner, however well realized, is not necessary for the effective functioning of the learning environment (see, e.g., VanLehn et al., 2005).

USABILITY CHALLENGES

Some of the typical properties of user-adaptive systems can lead to usability problems that may outweigh the benefits of adaptation to the individual user. Discussions of these problems have been presented by a number of authors (e.g., Norman, 1994; Wexelblat & Maes, 1997; Höök, 2000; Tsandilas & Schraefel, 2004; and the references given below in this section). Fig. 22.23 gives a high-level summary of many of the relevant ideas, using

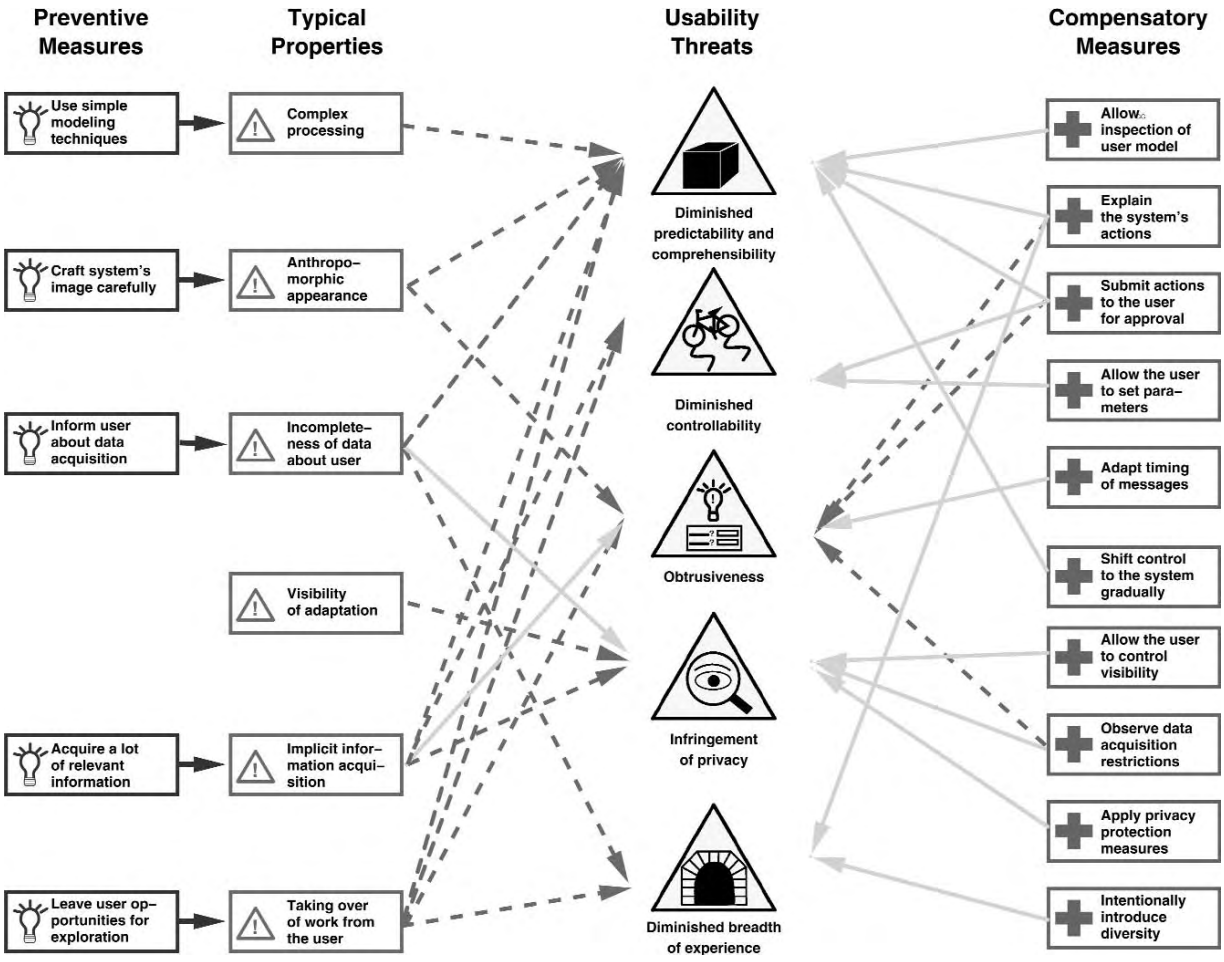


FIGURE 22.23. Overview of usability challenges for user-adaptive systems and of ways of dealing with them. (Dashed arrows denote threats and solid arrows mitigation of threats, respectively; further explanation is given in the text.)

the metaphor of signs that give warnings and advice to persons who enter a dangerous area.

The Usability Threats, shown in the third column, concern several generally desirable properties of interactive systems. Those referred to by the top three signs (concerning predictability and comprehensibility, controllability, and unobtrusiveness) correspond to general usability principles. The remaining two threats, to privacy and to breadth of experience are especially relevant to user-adaptive systems.

The Typical Properties column lists some frequently encountered (though not always necessary) properties of user-adaptive systems, each of which has the potential of creating particular usability threats.

Each of the remaining two columns shows a different strategy for avoiding or mitigating one or more usability threats: Each of the Preventive Measures aims to ensure that one of the Typical Properties is not present in such a way that it would cause problems. Each of the Compensatory Measures aims to ward off one or more threats once it has arisen.

A discussion of all of the relationships indicated in Fig. 22.23 would exceed the scope of this chapter, but some remarks will clarify the main ideas.

Threats to Predictability and Comprehensibility

The concept of *predictability* refers to the extent to which a user can predict the effects of his or her actions. *Comprehensibility* is the extent to which the user can understand system actions and/or has a clear picture of how the system works.[14] These goals are grouped together here because they are associated with largely the same set of other variables.

Users can try to predict and understand a system on several different levels of detail.

1. *Exact layout and responses.* Especially detailed predictability is important when interface elements are involved that are accessed frequently by skilled users—for example, icons in control panels or options in menus (cf. the discussion of interface adaptations above). If the layout and behavior of the system is highly predictable—in fact, essentially identical—over time, skilled users can engage in *automatic processing* (see, e.g., Hammond, 1987): They can use the parts of the interface quickly, accurately, and with little or no attention. In this situation, even minor deviations from complete predictability on a fine-grained level can have the serious consequence of making automatic processing impossible or error-prone.
2. *Success at specific subtasks.* Users may desire only more global predictability and comprehensibility when the system is performing some more or less complex task on the user's behalf (e.g., searching for suitable products on the web): In the extreme case, the user may want only to predict (or evaluate) the quality of the result of a complex system action.
3. *Overall competence.* The most global form of predictability and comprehensibility concerns the user's ability to assess the system's overall level of competence: the degree to which the system tends in general to perform its tasks successfully. With many types of system, high overall competence can be taken for granted; but as we have seen, the processes of acquiring and applying user models do not in general ensure a high degree of accuracy. If the user seriously overestimates the system's competence, he or she may rely on the system excessively; if the user underestimates the system, he or she will not derive the potential benefits that the system can provide. A factor that is especially important with regard to this global level is the way in which the adaptive part of the system is presented to the user. Some user-adaptive systems, such as AgentSalon (which was discussed above) and the well-known Microsoft Office Assistant, have employed lifelike characters, for various reasons. As has often been pointed out, such anthropomorphic representations can invoke unrealistically high expectations concerning system competence—not only with regard to capabilities like natural language understanding but also with regard to the system's ability to understand and adapt to the user.

In general, the levels and degrees of predictability and comprehensibility that are necessary or desirable in a given case can depend on many factors, including the function that is being served by the adaptation and the user's level of skill and experience. The same is true of the choice of the strategies that are most appropriate for the achievement of predictability and comprehensibility.

Threats to Controllability

Controllability refers to the extent to which the user can bring about or prevent particular actions or states of the system if he or she has the goal of doing so. Although controllability tends to be enhanced by comprehensibility and predictability, these properties are not perfectly correlated. For example, when the user clicks on a previously unused option in Smart Menus, he or she can predict with certainty that it will be moved to the main part of its menu; but the user has no control over whether this change will be made.

A typical measure for ensuring some degree of control is to have the system submit any action with significant consequences to the user for approval. This measure cause a threat of obtrusiveness (see below); so it is an important interface design challenge to find ways of making recommendations in an unobtrusive fashion that still makes it easy for the user to notice and follow up on them (cf. the earlier discussion of FXPAL Bar).

Like predictability and comprehensibility, controllability can be achieved on various levels of granularity. Especially since the enhancement of controllability can come at a price, it is important to consider what kinds of control will really be desired. For example, there may be little point in submitting individual actions to the user for approval if the user lacks the knowledge or interest required to make the decisions. Wexelblat and Maes (1997) recommended making available several alternative types of control from which users can choose.

[14]The term *transparency* is sometimes used for this concept, but it can be confusing, because it also has different, incompatible meanings.

Obtrusiveness

We will use the term *obtrusiveness* to refer to the extent to which the system places demands on the user's attention which reduce the user's ability to concentrate on her primary tasks. This term—and the related words *distracting* and *irritating*—are often heard in connection with user-adaptive systems. Fig. 22.23 shows that (a) there are several different reasons why user-adaptive systems can easily turn out to be obtrusive and (b) there are equally many corresponding strategies for minimizing obtrusiveness. Some of these measures can lead straightforwardly to significant improvements—for example, when it is recognized that distracting lifelike behaviors of an animated character are not really a necessary part of the system.

Threats to Privacy

User-adaptive systems typically (a) gather data about individual users and (b) use these data to make decisions that may have more or less serious consequences. Users may accordingly become concerned about the possibility that their data will be put to inappropriate use. Privacy concerns tend to be especially acute in e-commerce contexts (e.g., Cranor, 2004) and with some forms of support for collaboration (e.g., Terveen & McDonald, 2004), because in these cases (a) data about the user are typically stored on computers other than the user's own, (b) the data often include personally identifying information, and (c) there may be strong incentives to use the data in ways that are not dictated by the user's own interests. As will be discussed in the next section, different means of acquiring information about users can have different consequences with regard to privacy. On the other hand, many of the measures that can be taken to protect privacy—for example, a policy of storing as little personally identifying data as possible—are not specific to user-adaptive systems.

Breadth of Experience

When a user-adaptive system helps the user with some form of information acquisition (cf. the second major section of this chapter) much of the work of examining the individual documents, products, and people involved is typically taken over by the system. A consequence can be that the user ends up learning less about the domain in question than she would with a nonadaptive system (cf. Lanier, 1995). For example, if the Amazon.com visitor for whom recommendations are shown in Fig. 22.1 relies heavily on such recommendations (as opposed to browsing freely), the user is likely to learn a lot about the books of Frederick Forsyth and about closely related products but little about the full range of books and other media that are available. One point of view here (see, e.g., the remarks of Maes in Shneiderman & Maes, 1997, p. 53) is that it should be up to the user to decide whether he or she prefers to learn about a given domain or to save time by delegating work to a system. It may be worthwhile to give the user a continuous spectrum of possibilities between complete control over a task and complete delegation of it. For example, many product recommendation systems, such as Amazon.com's, allow users to alternate freely between pursuing the system's recommendations and browsing through product descriptions in the normal way.

Reduction of breadth of experience is especially likely if the system relies more heavily than is necessary on an incomplete user model. The user of Fig. 22.1 had (understandably) informed the system about only a tiny proportion of the books that he had ever read and liked; and since the system tends to recommend—and offer a chance to rate—similar items, the system may never obtain much evidence about all of the user's other literary interests. Some systems mitigate this problem by systematically proposing solutions that are not dictated by the current user model (for a method that is directly applicable to recommendation lists such as Amazon.com's, see Ziegler, McNee, Konstan, & Lausen, 2005; for methods realized in different types of recommenders, see Linden, Hanks, & Lesh, 1997; Shearin & Lieberman, 2001).

Dealing With Tradeoffs

As can be seen in Fig. 22.23, the designer who attempts to combat a particular usability threat will often have to accept a greater threat to some other usability goal. The most obvious tradeoffs involve unobstrusiveness. In particular, steps taken to enhance control or to protect privacy often require the user to perform additional actions, input additional information, and pay attention to additional system messages. Dealing with tradeoffs of this sort is complicated by the fact that users often differ markedly in the relative priority that they assign to each of the conflicting goals.

Fig. 22.24 illustrates some general points, referring for concreteness to a recently developed prototype Office Control

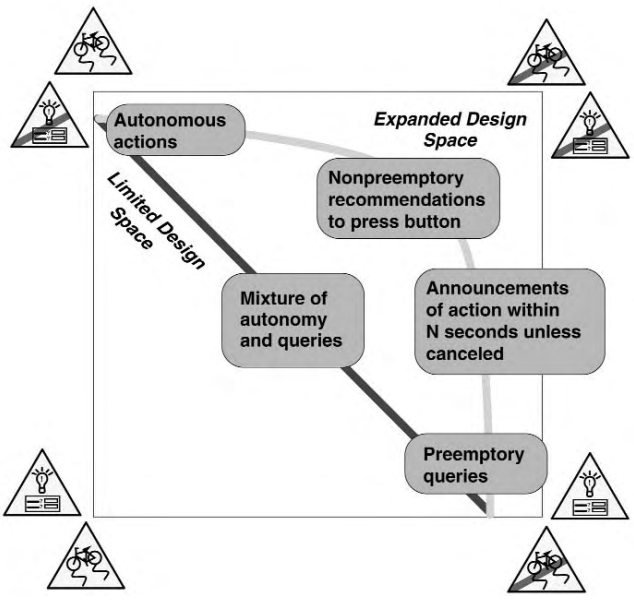


FIGURE 22.24. Illustration of strategies for dealing with tradeoffs among usability goals in user-adaptive systems. (Explanation in text.)

System (Cheverst et al., 2005). This system first observes how the occupant of an office tends to operate various devices such as the fan and the window shades; it then tries to help the user by performing some actions autonomously (e.g., opening the window when certain weather conditions prevail and there is no visitor in the office). An early version offered two ways of dividing the work between the user and the system for each type of action: The system could either perform actions of that type autonomously or request the user's permission with a preemptory dialogue box on the user's normal computer screen. Users chose the latter option for different proportions of the available actions, reflecting different priorities for the goals of unobtrusiveness and controllability respectively. In terms of the tradeoff graph shown in Fig. 22.24, users chose different points on the straight diagonal line. Despite this freedom of choice, users were often not satisfied with the overall usability of the prototype. A significant improvement in acceptance was achieved when the designers expanded the design space: They introduced a separate small screen for the Office Control System, in which information and requests for confirmation could be offered in several ways that are not very familiar in everyday graphical user interfaces (though they are familiar in industrial and traffic contexts; see, e.g., Wickens & Hollands, 2000). As Fig. 22.24 indicates, these additional forms of interaction represented a more favorable combination of degrees of unobtrusiveness and controllability at least for some of the users some of the time.

Consistent with more complex examples (see Jameson & Schwarzkopf, 2002; Billsus & Pazzani, 2007), this small case study illustrates two general points: (a) When dealing with tradeoffs among the usability goals discussed here, it is important to offer alternative solutions for users with different priorities. (b) It is equally important to consider relatively novel interface design solutions that may spare users the need to choose among unsatisfactory alternatives.

OBTAINING INFORMATION ABOUT USERS

Some of the usability challenges discussed in the previous section are closely connected with the ways in which information about individual users is acquired—a consideration which also largely determines the success of a system's adaptation. The next two subsections will look, respectively, at (a) information that the user supplies to the system explicitly for the purpose of allowing the system to adapt and (b) information that the system obtains in some other way.

Explicit Self-Reports and -Assessments

Self-reports about objective personal characteristics. Information about objective properties of the user (such as age, profession, and place of residence) sometimes has implications that are relevant for system adaptation—for example, concerning the topics in which the user is likely to be knowledgeable or interested. This type of information also has the advantage of changing relatively infrequently. Some user-adaptive systems request information of this type from users, but the following caveats apply:

1. Specifying information such as profession and place of residence may require a fair amount of tedious menu selection and/or typing.
2. Since information of this sort can often be used to determine the user's identity, the user may justifiably be concerned about privacy issues. Even in cases where such concerns are unfounded, they may discourage the user from entering the requested information.

A general approach is (a) to restrict requests for personal data to the few pieces of information (if any) that the system really requires and (b) to explain the uses to which the data will be put. Cranor (2004) gave a number of suggestions about how the use of personally identifying data can be minimized. An especially creative approach was tried in the web-based Lifestyle Finder prototype (Fig. 22.25; Krulwich, 1997), which was characterized by a playful style and an absence of requests for personally identifying information. Of the users surveyed, 93% agreed that the Lifestyle Finder's questions did not invade their privacy.

It is sometimes possible to avoid requests for explicit input about personal characteristics by accessing sources where similar information has already been stored (a strategy that will be discussed later in this section).

Self-assessments of interests and knowledge. It is sometimes helpful for a user-adaptive system to have an assessment of a property of the user that can be expressed naturally as a position on a particular general dimension: the level of the user's interest in a particular topic, the level of his or her knowledge about it, or the importance that the user attaches to a particular evaluation criterion. Often an assessment is arrived at through inference on the basis of indirect evidence, as with the assessments of a learner's knowledge in the SQL-Tutor (Fig. 22.21). But it may be necessary or more efficient to ask the user

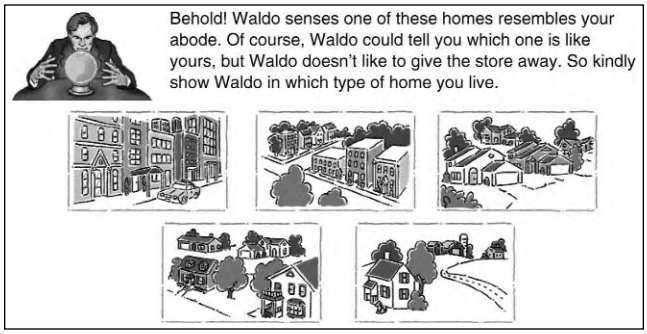


FIGURE 22.25. Example of a screen with which the LifeStyle Finder elicits demographic information. (Figure 3 of "Lifestyle Finder: Intelligent user profiling using large-scale demographic data," by B. Krulwich, 1997, AI M*agazine*, 18(2), pp. 37–45. Research conducted at the Center for Strategic Technology Research of Andersen Consulting (now Accenture Technology Labs). Copyright 1997 by the American Association for Artificial Intelligence. Adapted with permission.)

for an explicit assessment. For example, it would be difficult for the Active Buyers Guide recommender shown in Fig. 22.16 to estimate the importance that the user attaches to photo quality without asking the user directly. The scales in this figure illustrate good practice in that they make clear the meaning of the various possible answers, instead of asking, "How important is photo quality to you (on a scale from 1 to 5)?"

Because of the effort involved in this type of self-assessment, it is generally worthwhile to consider ways of minimizing such requests, making responses optional, ensuring that the purpose is clear, and integrating the self-assessment process into the user's main task (for some innovative ideas about how to achieve these goals, see Tsandilas & schraefel, 2004).

Self-reports on specific evaluations. Instead of asking the user to describe his or her interests explicitly, some systems try to infer the user's position on the basis of his or her explicitly evaluative responses to specific items. Amazon.com's rating scales (Fig. 22.1) illustrate one form that this type of input can take; other forms include checkboxes and icons (e.g., "thumbs-up" and "thumbs-down"). The items that the user evaluates can be (a) items that the user is currently experiencing directly (e.g., the current website); (b) actions that the system has just performed, which the user may want to encourage or discourage (see, e.g., Wolfman, Lau, Domingos, & Weld, 2001); (c) items that the user must judge on the basis of a description (e.g., the abstract of a talk; a table listing the attributes of a physical product); or (d) the mere name of an item (e.g., a movie) that the user may have had some experience with in the past (see, e.g., Fig. 22.1). The cognitive effort required depends in part on how directly available the item is: In the third and fourth cases just listed, the user may need to perform memory retrieval and inference in order to arrive at an evaluation.

Even when the effort is minimal, users often do not like to bother with explicit evaluations that do not constitute a necessary part of the task they are performing. For this reason, many designers try to get by with the type of nonexplicit input discussed in the following section. For example, Findory (Fig. 22.14) could allow the user to rate the various news stories and blogs presented, but instead it just interprets the user's behavior in selecting items to read them.

Responses to test items. In systems that support learning, it is often natural to administer tests of knowledge or skill. In addition to serving their normal educational functions, these tests can yield valuable information for the system's adaptation to the user. An advantage of tests is that they can be constructed, administered, and interpreted with the help of a large body of theory, methodology, and practical experience (e.g., Wainer, 2000).

Outside of a learning context, users are likely to hesitate to invest time in tests of knowledge or skill, unless these can be presented in an enjoyable form (e.g., the color discrimination test used by Gutkauf, Thies, & Domik, 1997, to identify perceptual limitations relevant to the automatic generation of graphs). Trewin (2004, p. 76) reported experience with a brief typing test that was designed to identify helpful keyboard adaptations: Some users who turned out to require no adaptations were disappointed that their investment in the test had yielded no benefit. As a result, Trewin decided that adaptations should be based on the users' naturally occurring typing behavior.

Nonexplicit Input

The previous subsection has given some examples of why designers often look for ways of obtaining information about the user that does not require any explicit input by the user.

Naturally Occurring Actions

The broadest and most important category of information of this type includes all of the actions that the user performs with the system that do not have the express purpose of revealing information about the user to the system. These actions may range from major actions like purchasing an expensive product to minor ones like scrolling down a web page. The more significant actions tend to be specific to the particular type of system that is involved (e.g., e-commerce websites vs. learning environments). Within some domains, there has been considerable research on ways of interpreting particular types of naturally occurring user actions. For example, researchers interested in adaptive hypertext navigation support have developed a variety of ways of analyzing the user's navigation actions to infer the user's interests and to propose navigation shortcuts (e.g., Mobasher, 2007).

In their purest forms, naturally occurring actions require no additional investment by the user, because they are actions that the user would perform anyway. The main limitation is that they are hard to interpret; for example, the fact that a given web page has been displayed in the user's browser for four minutes does not reveal with certainty which (if any) of the text displayed on that page the user has actually read. Some designers have tried to deal with this trade-off by designing the user interface in such a way that the naturally occurring actions are especially easy to interpret. For example, a web-based system might display just one news story on each page, even if displaying several stories on each page would normally be more desirable.

The interpretation of naturally occurring actions by the system can raise privacy and comprehensibility issues (cf. Fig. 22.23) that do not arise in the same way with explicit self-reports and self-assessments of the types discussed earlier in this section: Whereas the latter way of obtaining information about the user can be compared with interviewing, the former way is more like eavesdropping—unless the user is informed about the nature of the data that are being collected and the ways in which they will be used (cf. Cranor, 2004).

Previously Stored Information

Sometimes a system can access relevant information about the user that has been acquired and stored independently of the system's interaction with the user:

1. If the user has some relationship (e.g., patient, customer) with the organization that operates the system, this organi-

zation may have information about the user that it has stored for reasons unrelated to any adaptation, such as the user's medical record (see Cawsey et al., 2007, for examples) or address.
2. Relevant information about the user may be stored in publicly available sources such as electronic directories or web home pages. For example, Pazzani (1999) explored the idea of using a user's web home page as a source of information for a restaurant recommending system.
3. If there is some other system that has already built up a model of the user, the system may be able to access the results of that modeling effort and try to apply them to its own modeling task. There is a line of research that deals with *user modeling servers* (e.g., Kobsa, 2007): systems that store information about users centrally and supply such information to a number of different applications. Some of the major commercial personalization software is based on this conception.

Relative to all of the other types of information about users, previously stored information has the advantage that it can be applied in principle right from the start of the first interaction of a given user with a given system. To be sure, the interpretability and usefulness of the information in the context of the current application may be limited. Moreover, questions concerning privacy and comprehensibility may be even more important than with the interpretation of naturally occurring actions.

Low-Level Indices of Psychological States

The next two categories of information about the user have become practically feasible only in recent years, with advances in the miniaturization of sensing devices.

The first category of sensor-based information (discussed at length in the classic book of Picard, 1997) comprises data that reflect aspects of a user's psychological state. Some of the application scenarios in which this type of information can be useful were discussed in the section on systems that mediate interaction with the real world.

Two categories of sensing devices have been employed: (a) devices attached to the user's body (or to the computing device itself) that transmit physiological data, such as electromyogram signals, the galvanic skin response, blood volume pressure, and the pattern of respiration (for an overview, see Lisetti & Nasoz, 2004); and (b) video cameras and microphones that transmit psychologically relevant information about the user, such as features of his or her facial expressions (e.g., Bartlett, Littlewort, Fasel, & Movellan, 2003) or his or her speech (e.g., Liscombe et al., 2005).

With both categories of sensors, the extraction of meaningful features from the low-level data stream requires the application of pattern recognition techniques. These typically make use of the results of machine learning studies in which the relationships between low-level data and meaningful features have been learned.

One advantage of sensors is that they supply a continuous stream of data, the cost to the user being limited to the physical and social discomfort that may be associated with the carrying or wearing of the devices. These factors are still significant now, but further advances in miniaturization—and perhaps changing attitudes as well—seem likely to reduce their importance.

Signals Concerning the Current Surroundings

As computing devices become more portable, it is becoming increasingly important for a user-adaptive system to have information about the user's current surroundings. Here again, two broad categories of input devices can be distinguished (for a discussion of a number of specific types of devices, see Krüger, Baus, Heckmann, Kruppa, & Wasinger, 2007).

1. Devices that receive explicit signals about the user's surroundings from specialized transmitters. Some mobile systems that are used outdoors employ GPS (Global Positioning System) technology. More specialized transmitters and receivers are required, for example, if a portable museum guide system is to be able to determine which exhibit the user is looking at.
2. More general sensing or input devices. For example, Schiele, Starner, Rhodes, Clarkson, and Pentland (2000) described the use of a miniature video camera and microphone (each roughly the size of a coin) that enable a wearable computer to discriminate among different types of surroundings (e.g., a supermarket vs. a street). The use of general-purpose sensors eliminates the dependence on specialized transmitters. On the other hand, the interpretation of the signals requires the use of sophisticated machine learning and pattern recognition techniques.

SPECIAL CONSIDERATIONS CONCERNING EMPIRICAL METHODS

The full repertoire of empirical methods in HCI is in principle applicable to user-adaptive systems. This section will focus on some methods that are more important for user-adaptive systems than for other types and on some typical problems that need to be dealt with. This focused discussion should not obscure the fact that a lot of empirical work with user-adaptive systems looks the same as with other systems.[15]

Use of Data Collected With a Nonadaptive System

The key difference between user-adaptive systems and other interactive systems is the inclusion of some method for acquiring and exploiting a user model. This feature gives rise to a type of empirical study that is largely unique to user-adaptive

[15]More extended discussions of empirical methods for user-adaptive systems are provided by Gena and Weibelzahl (2007), Höök (2000), and Langley and Fehling (1998).

systems: studies in which the accuracy of the modeling methods is evaluated.

This type of evaluation can often be performed even if no user-adaptive systems that employ the user modeling method in question exist. What is needed are (a) some implementation of the adaptation algorithm, not necessarily embedded in any interactive system, and (b) a database of behavioral data on how a number of users have used a relevant nonadaptive system. The researcher can then apply the modeling method to the data in order to determine how well the system would adapt to the users in question.

A number of studies of this type were conducted with the i-ems system (see McCreath et al., 2006, and the discussion earlier in this chapter). In one case, the researchers wanted to find out whether a user who had defined a number of handcrafted e-mail sorting rules could benefit from having automatically learned rules applied to messages that were not covered by the handcrafted rules. One simulation was performed on 5,100 e-mail messages that had been sorted by a single user within a nonadaptive e-mail client over three months. In the order in which the messages had been received, they were presented to one of the system's learning algorithms in batches of 100, along with information about how the user had sorted them, so that the system could continually refine its set of learned rules. After each batch of 100 messages, the accuracy of the updated set of rules was evaluated: The system was asked to make predictions about the next batch of messages before being told how the user had in fact sorted them, and these predictions were compared with the user's actual behavior. Several indices of the system's performance were computed, one of which is shown in Fig. 22.26: the percentage of messages for which the learned rules made no prediction (e.g., where the appropriate folder was unknown to the system). The middle curve in the graph shows how this percentage gradually decreased over time. The uppermost curve shows the corresponding results for the case where only the user's handcrafted rules were used for prediction; the lowest curve shows the results where both types of rules were applied, with the handcrafted rules taking precedence in the case of disagreement. It can be seen that the joint use of both sets of rules gave the best results in terms of avoiding "unknown" predictions. Since according to other indices the accuracy of the combined set of rules was as good as that of the handcrafted rules alone, the system's performance was best overall with the combined set.

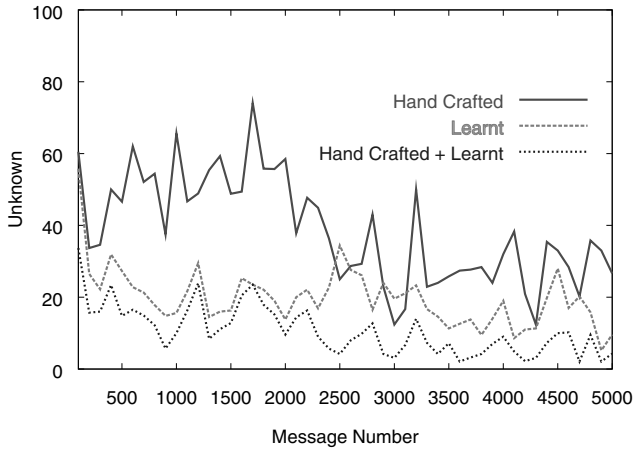


FIGURE 22.26. Results of one of the simulation experiments performed with i-ems's algorithms. (Explanation in text. Figure courtesy of Eric McCreath.)

Note that it was not necessary, for this evaluation, to create three different versions of i-ems and have them used for months by different users. In addition to being extremely time-consuming, this procedure would allow less direct accuracy comparisons. By contrast, using existing e-mail corpora, McCreath et al. (2006) performed simulations that shed light on many properties of the algorithms used.

The appeal of this type of evaluation, in terms of being able to yield numerous interpretable results with minimal involvement of actual users, is so great that researchers often seem to lose sight of the fact that studies with real users are likewise essential. No simulation study, for example, can reveal how well the design of the i-ems interface shown in Fig. 22.4 corresponds to the way users like to deal with incoming e-mail, or how helpful users find the explanations that the system offers for its predictions.

Early Studies of Usage Scenarios and User Requirements

In the field of HCI as a whole, it is expected that user-centered design should begin with a study of contexts of use, usage scenarios, properties of users, and user requirements. To date, this strategy has been applied less frequently in the design of user-adaptive systems—perhaps because the designers less frequently come from an HCI background, often specializing instead in the development of the necessary adaptive algorithms. Early user studies are actually at least as important with novel user-adaptive systems as with other types of system: It is often not clear in advance whether adaptation will yield added value and achieve acceptance in a particular context. Careful attention to the requirements and contexts of users may greatly increase the likelihood of success—or at least warn of the designers at an early stage that if a particular usage scenario is not promising for the sort of adaptive interaction that they envision.

A positive example of early attention to user requirements is found in the development of the museum guide HyperAudio, which was developed as a prototype in the 1990s (for a retrospective discussion, see Petrelli & Not, 2005). Studying the attitudes and behavior of museum visitors at an early stage in the system's design, the researchers found that many visitors enjoy guided tours but that few visitors want to spend time interacting with technical devices. These two findings, along with others, led to a modification of the original conception of HyperAudio, in that they suggested the appropriateness of a museum guide that selects information for presentation on the basis of the user's behavior and location, requiring little or no explicit input.

Wizard of Oz Studies

Systems that adapt to their users are in one methodological respect similar to systems that make use of speech: They attempt to realize a capability that is so far, at least in many contexts,

possessed to the highest degree by humans. Consequently, as with speech interfaces, valuable information can sometimes be obtained from a Wizard of Oz study: In a specially created setting, a human takes over a part of the processing of the to-be-developed system for which humans are especially well suited.

One example is the Wizard of Oz study that was conducted early in the development of the Lumière intelligent help system (Horvitz, Breese, Heckerman, Hovel, & Rommelse, 1998), which formed the basis for the Office Assistant, which was introduced in Microsoft Office 97. In this study, subjects working with a spreadsheet were told that an experimental help system would track their activities and make guesses about how to help them. They received the advice via computer monitors. The advice was actually provided by usability experts who, working in a separate room, viewed the subjects' activity via monitors and conveyed their advice by typing.

This type of study can yield an upper-bound estimate of the highest level of modeling accuracy that might be attainable given the available information—as long as one can assume that the human wizards are more competent at the type of assessment in question than a fully automatic system is likely to be in the foreseeable future. In this example study, the expert advisers showed some ability to identify the users' goals and needs; if they had not done so, perhaps the entire project would have been reconsidered.

Aside from accuracy, a Wizard of Oz study can also shed light on problems with the acceptability and usability of the new system, as long as they concern the content and basic nature of the adaptations performed, as opposed to interface details that are not faithfully reproduced in the study. In our example study, it turned out that even incorrect advice was often taken seriously by the users, who wasted time following up on irrelevant suggestions. One design implication is that users should be made aware of the fact that the system's advice is not necessarily relevant.

Comparisons With the Work of Human Designers

Just as humans can sometimes be employed as a surrogate for a user-adaptive system in the early stages of design, humans can sometimes also serve as a standard of comparison for the evaluation of an implemented system. This method makes most sense when the system is performing a task at which human authors or designers are likely to be experienced and skilled, such as the tailoring of the content of a presentation to the individual user. An example of such a study was mentioned earlier: a comparison of the displays generated by the real estate recommender system RIA with those generated by two experienced designers. Instead of performing the same task as the system, the human designers may simply act as judges of the appropriateness of the system's output. In either case, the comments of the designers can yield valuable qualitative information to complement the objective results.

Experimental Comparisons of Adaptive and Nonadaptive Systems

Many experimental studies involving user-adaptive systems compare an adaptive variant with some nonadaptive one. This

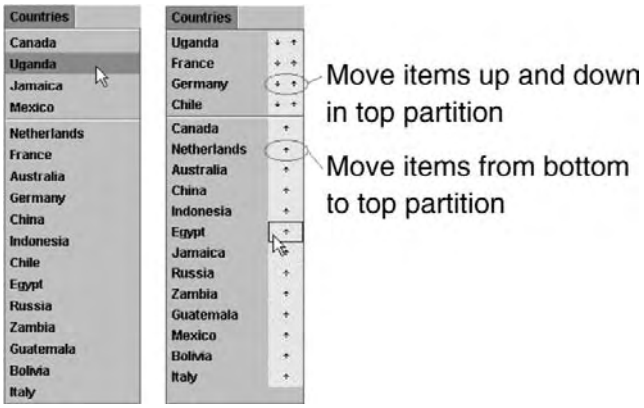


FIGURE 22.27. The adaptive (left) and adaptable (right) split menus used in an experimental comparison. (Adapted from Figure 1 of "A comparison of static, adaptive, and adaptable menus," by L. Findlater & J. McGrenere, 2004, in E. Dykstra-Erickson & M. Tscheligi (Eds.), H*uman factors in computing systems*: CHI 2004 *conference proceedings,* pp. 89–96, New York: ACM, Copyright 2004 by the Association for Computing Machinery, Inc. Adapted with permission.)

strategy is understandable given the doubt that often exists as to whether the additional overhead required for user-adaptivity is justified by any sort of improvement in the interaction. However, studies like this are trickier to conduct and interpret than they may seem at first glance.

As an example, consider the experiment by Findlater and McGrenere (2004), which examined whether subjects could work faster using (a) an adaptive menu system somewhat like Smart Menus (cf. the discussion of adaptive menus above); (b) an adaptable system in which the users explicitly determined the content of the menus themselves; or (c) a conventional static menu system. For each of these three types of menu, a realization was chosen that seemed optimal in the context of the experiment. All three menus were a special type of split menu (Sears & Shneiderman, 1994): The four most frequent items in each menu were placed in a special section at the top of the menu for quick access (instead of being temporarily hidden, as in Smart Menus). The items selected for the upper part of the static menu were optimally chosen in that they reflected the actual frequency of the items in the experimental tasks. The adaptive menu was initially identical to the static menu, but the arrangement of the items changed as a function of the user's behavior, favoring the most frequently and recently used menu items. For the adaptable menu, the upper part was initially empty, so that users would be encouraged to perform some adaptations. Though the overall pattern of results is complex, it tends to speak in favor of the adaptable menu. Note that the conditions did not give the adaptive variant much of a chance to provide any benefit. Since the initial menu was already the best possible single menu for the experimental tasks, adaptation could improve performance only by taking advantage of any local concentrations of commands within a particular period of time (e.g., the need to execute the same command several times in succession). By contrast, in normal usage situations, an adaptive menu can also improve

performance by reflecting increasingly the user's longer-term patterns of use.

The difficulties in interpreting the results of this experiment could not easily have been avoided with a different design: Any other way of realizing the three conditions would have left some different set of questions open. The lesson of this and many other examples is that comparisons between adaptive and nonadaptive variants of a system should not be viewed as empirical tests whose results can be interpreted straightforwardly. Instead, they should be seen as shedding light on various aspects of the ways in which people use adaptive and nonadaptive systems and on the effectiveness of these methods in certain conditions.

Taking Into Account Individual Differences

Individual differences among users show up in just about every user study in the field of HCI; but with user-adaptive systems, they are especially important because of the wide range of subjective reactions that user-adaptivity tends to invoke (illustrated in the experiment discussed in the previous subsection). As a result, asking whether people like a particular type of user-adaptive system is in many cases like asking whether the voters in a given country prefer progressive or conservative policies. Even if, in a given sample, a statistically significant tendency in one direction or the other can be found, important minority points of view should be understood and reported. As in politics, the goal should be to take into account the range of different preferences in a way that is satisfactory to at least a large proportion of the potential user group.

When individual differences are present, it may be tempting to try to find correlations with demographic characteristics or with general personality variables. Although some relationships of this sort can be found (see, for example, Graziola, Pianesi, Zancanaro, & Goren-Bar, 2005, with regard to personality variables), it is not always worthwhile to look for them. The relationships tend to be weak, since different responses can also be due to more specific causes, such as degree of familiarity with the particular type of system and particular conditions under which a user performs a task.

Checking Usability Under Realistic Conditions

With just about any type of interactive system, new lessons are likely to be learned when a working prototype (or the finished system itself) is tested in realistic situations, even if the system has been studied thoroughly in earlier stages. With user-adaptive systems, realistic testing is especially advisable because of the issues discussed in the section on usability challenges, whose importance for a given system can often be assessed only in real use. For example, an obtrusive proactive recommender might be considered quite acceptable, or even amusing, when the user is performing some artificial assigned task in a laboratory; in the real world, when he or she is under pressure to complete an important task quickly, the interruptions may be evaluated quite differently. Similarly, privacy issues are serious mainly when real data about the user are involved and when misuse could have real consequences.

THE FUTURE OF USER-ADAPTIVE SYSTEMS

This chapter has shown that adaptive interfaces, agents, and other user-adaptive systems do not represent a smooth and easy shortcut to more successful HCI: They present a complex set of usability challenges, and they require carefully designed methods of acquiring information about users, as well as relatively sophisticated computational techniques that are not needed in other types of interactive system. Even when all of these requirements have been dealt with, it is often tricky to prove empirically that user-adaptivity has actually added any value. It is no wonder that some experts believe that the interests of computer users are better served by continued progress within more familiar paradigms of user-centered system design.

On the other hand, our understanding of the complex challenges raised by user-adaptive systems has been growing steadily, and they are now familiar and valued elements in a number of types of system, as the survey in the first two major sections of this chapter showed.

Growing Need for User-Adaptivity

Increases in the following variables suggest that the functions served by user-adaptivity will continue to grow in importance:

Diversity of users and contexts of use. As several chapters in this handbook make clear, computing devices are being used by an ever-increasing variety of users in an increasing variety of contexts. It is therefore becoming harder to design a system that will be suitable for all users and contexts without some sort of user-adaptivity or user-controlled adaptability; and as has been discussed at several points in this chapter, user-controlled adaptability has its limitations.

Number and complexity of interactive systems. The functions of user-adaptivity discussed in the first major section of this chapter involve helping users to deal effectively with interactive systems and tasks even when they are not able or willing to gain complete understanding and control in each individual case. This goal becomes increasingly important as the number—and in some cases the complexity—of the systems that people must deal with continues to increase—because of factors ranging from the growth of the World Wide Web to the proliferation of miniature interactive computing devices.

Scope of information to be dealt with. Even when using a single, relatively simple system, users today can often access a much larger and more diverse set of objects of interest than they could a few years ago—be they documents, products, or potential collaborators. It is therefore becoming relatively more attractive to delegate some of the work of dealing with these objects—even to a system that has an imperfect model of the user's requirements. In the early 1990s, the idea that an e-mail sorting agent, such as the one described by Maes (1994), might delete an incoming message without consulting the user seemed preposterous to many people. After the huge increase in the amount of (largely unwanted) e-mail that has occurred

since then, many people now regularly allow dozens of their incoming messages to be deleted unseen.

Increasing Feasibility of Successful Adaptation

As the need for user-adaptivity increases, fortunately so does its feasibility, largely because of advances in the following areas:

Ways of acquiring information about users. Most of the methods discussed in the section about acquiring information about users are becoming more powerful with advances in technology and research. They therefore offer the prospect of substantial increases in the quality of adaptation—although methods for ensuring users' privacy call for equal attention.

Advances in techniques for learning, inference, and decision. In addition to the more general progress in the fields of machine learning and artificial intelligence, communities of researchers have been focusing on the specific requirements of computational techniques that support user-adaptivity. Consequently, noticeable progress is being made every year.

Attention to empirical methods. The special empirical issues and methods that are involved in the design and evaluation of user-adaptive systems have been receiving increasing attention from researchers, as emphasis has shifted from high technical sophistication to ensuring that the systems enhance the users' experience.

Despite these tendencies, it is actually unlikely that the number of deployed systems associated with labels like "user-adaptive" will increase. Once an adaptation technique has left the research laboratory and started playing some genuinely useful role in people's lives, it tends to be described in terms of the function that it serves rather than in terms of the techniques that it uses. Awareness of the commonalities discussed in this chapter should help both to increase the number of systems that succeed in this way and to recognize them despite the new labels that are placed on them.

ACKNOWLEDGMENTS

Preparation of this chapter was supported by the German Ministry of Education and Research (projects Specter and Collate). The following colleagues provided valuable assistance in the form of comments on an earlier draft, supplementary information about their research, and/or graphics depicting their systems: Vikram Aggarwal and Michelle Zhou; James 'Bo' Begole and Nicholas Matsakis; Vania Dimitrova; Tatsuya Kawahara; Bruce Krulwich; Greg Linden; Eric McCreath and Judy Kay; Joanna McGrenere and Leah Findlater; Antonija Mitrovic; Charles Rich and Elyon DeKoven; Yasuyuki Sumi and Kenji Mase; Karin Vogel.

References

Aberg, J., & Shahmehri, N. (2001). An empirical study of human web assistants: Implications for user support in web information systems. In J. A. Jacko, A. Sears, M. Beaudouin-Lafon, & R. J. Jacob (Eds.), *Human factors in computing systems: CHI 2001 conference proceedings* (pp. 400–411). New York: ACM.

Bälter, O., & Sidner, C. L. (2002). Bifrost Inbox Organizer: Giving users control over the inbox. *Proceedings of NORDICHI 2002*, 111–118.

Bartlett, M. S., Littlewort, G., Fasel, I., & Movellan, J. R. (2003). Real time face detection and facial expression recognition: Development and applications to human computer interaction. *Proceedings of the Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction at the 2003 Conference on Computer Vision and Pattern Recognition*, 53.

Billsus, D., & Pazzani, M. (2007). Adaptive news access. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 550–572).

Billsus, D., Hilbert, D., & Maynes-Aminzade, D. (2005). Improving proactive information systems. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 159–166). New York: ACM.

Bontcheva, K., & Wilks, Y. (2005). Tailoring automatically generated hypertext. *User Modeling and User-Adapted Interaction, 15*, 135–168.

Brusilovsky, P., Kobsa, A., & Nejdl, W. (Eds.). (2007). *The adaptive web: Methods and strategies of web personalization.* Berlin: Springer.

Budzik, J., Hammond, K., & Birnbaum, L. (2001). Information access in context. *Knowledge-Based Systems, 14*, 37–53.

Bunt, A., Carenini, G., & Conati, C. (2007). Adaptive presentation for the web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 409–432). Berlin: Springer.

Bunt, A., Conati, C., & McGrenere, J. (2004). What role can adaptive support play in an adaptable system? In C. Rich & N. J. Nunes (Eds.), *IUI 2004: International Conference on Intelligent User Interfaces* (pp. 117–124). New York: ACM.

Burke, R. D., Hammond, K. J., & Young, B. C. (1997). The FindMe approach to assisted browsing. *IEEE Expert, 12*(4), 32–40.

Cawsey, A., Grasso, F., & Paris, C. (2007). Adaptive information for consumers of healthcare. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 465–484). Berlin: Springer.

Cheverst, K., Byun, H. E., Fitton, D., Sas, C., Kray, C., & Villar, N. (2005). Exploring issues of user model transparency and proactive behaviour in an office environment control system. *User Modeling and User-Adapted Interaction, 15*(3/4), 235–273.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. Gmytrasiewicz, & J. Vassileva (Eds.), *UM2001, User Modeling: Proceedings of the Eighth International Conference* (pp. 137–147). Berlin: Springer.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. In M. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 849–874). Amsterdam: North-Holland.

Cranor, L. F. (2004). 'I didn't buy it for myself': Privacy and e-commerce personalization. In C. Karat, J. Blom, & J. Karat (Eds.), *Designing*

personalized user experiences for eCommerce (pp. 57–74). Dordrecht, The Netherlands: Kluwer.

Crawford, E., Kay, J., & McCreath, E. (2002). An intelligent interface for sorting electronic mail. In Y. Gil & D. B. Leake (Eds.), *IUI 2002: International Conference on Intelligent User Interfaces* (pp. 182–183). New York: ACM.

DeKoven, E. A. (2004). *Help me help you: Designing support for person-product collaboration.* Unpublished doctoral dissertation, Delft University of Technology.

Fernandez, R., & Picard, R. W. (2000). Modeling drivers' speech under stress. *Proceedings of the ISCA Workshop on Speech and Emotions*, Belfast.

Findlater, L., & McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In E. Dykstra-Erickson & M. Tscheligi (Eds.), *Human factors in computing systems: CHI 2004 conference proceedings* (pp. 89–96). New York: ACM.

Fogarty, J., Ko, A. J., Aung, H. H., Golden, E., Tang, K. P., & Hudson, S. E. (2005). Examining task engagement in sensor-based statistical models of human interruptibility. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 331–340). New York: ACM.

Fuller, R. (2005). Towards a general theory of driver behaviour. *Accident Analysis & Prevention, 37*(3), 461–472.

Gena, C., & Weibelzahl, S. (2007. Usability engineering for the adoptive web. (2006). In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 720–766). Berlin: Springer.

Gervasio, M. T., Moffitt, M. D., Pollack, M. E., Taylor, J. M., & Uribe, T. E. (2005). Active preference learning for personalized calendar scheduling assistance. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 90–97). New York: ACM.

Graziola, I., Pianesi, F., Zancanaro, M., & Goren-Bar, D. (2005). Dimensions of adaptivity in mobile systems: Personality and people's attitudes. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 223–230). New York: ACM.

Gruen, D., Rohall, S. L., Minassian, S., Kerr, B., Moody, P., Stachel, B. B., et al. (2004). Lessons from the ReMail prototypes. In J. Herbsleb & G. Olson (Eds.), *Proceedings of the 2004 Conference on Computer-Supported Cooperative Work* (pp. 152–161). New York: ACM.

Gutkauf, B., Thies, S., & Domik, G. (1997). A user-adaptive chart editing system based on user modeling and critiquing. In A. Jameson, C. Paris, & C. Tasso (Eds.), *UM97 User modeling: Proceedings of the Sixth International Conference* (pp. 159–170). Vienna: Springer Wien New York.

Höök, K. (2000). Steps to take before IUIs become real. *Interacting with Computers, 12*(4), 409–426.

Haller, S., McRoy, S., & Kobsa, A. (Eds.). (1999). *Computational models of mixed-initiative interaction.* Dordrecht, The Netherlands: Kluwer.

Hammond, N. (1987). Principles from the psychology of skill acquisition. In M. M. Gardiner & B. Christie (Eds.), *Applying cognitive psychology to user interface design* (pp. 163–188). Chichester, England: Wiley.

Hanson, V. L., & Crayne, S. (2005). Personalization of web browsing: Adaptations to meet the needs of older adults. *Universal Access in the Information Society, 4*, 46–58.

Healey, J., & Picard, R. (2000). SmartCar: Detecting driver stress. *Proceedings of the Fifteenth International Conference on Pattern Recognition* (pp. 4218–4221), Barcelona.

Hegner, S. J., McKevitt, P., Norvig, P., & Wilensky, R. L. (Eds.). (2001). *Intelligent help systems for UNIX.* Dordrecht, The Netherlands: Kluwer.

Ho, J., & Intille, S. S. (2005). Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 909–918). New York: ACM.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In M. G. Williams, M. W. Altom, K. Ehrlich, & W. Newman (Eds.), *Human factors in computing systems: CHI '99 conference proceedings* (pp. 159–166). New York: ACM.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In G. F. Cooper & S. Moral (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference* (pp. 256–265). San Francisco: Morgan Kaufmann.

Horvitz, E., Koch, P., Sarin, R., Apacible, J., & Subramani, M. (2005). Bayesphone: Precomputation of context-sensitive policies for inquiry and action in mobile devices. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *UM2005, User Modeling: Proceedings of the Tenth International Conference* (pp. 251–260). Berlin: Springer.

Iqbal, S. T., Adamczyk, P. D., Zheng, X. S., & Bailey, B. P. (2005). Towards an index of opportunity: Understanding changes in mental workload during task execution. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 311–318). New York: ACM.

Jameson, A., & Schwarzkopf, E. (2002). Pros and cons of controllability: An empirical study. In P. De Bra, P. Brusilovsky, & R. Conejo (Eds.), *Adaptive hypermedia and adaptive web-based systems: Proceedings of AH 2002* (pp. 193–202). Berlin: Springer.

Jameson, A., & Smyth, B. (2007). Recommendation for groups. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 596–627). Berlin: Springer.

Jones, C. M., & Jonsson, I. (2005). Automatic recognition of affective cues in the speech of car drivers to allow appropriate responses. *Proceedings of OZCHI 2005*.

Kelly, D., & Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *ACM SIGIR Forum, 37*(2), 18–28.

Kobsa, A. (2007). Generic user modeling systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Berlin: Springer.

Kobsa, A., & Wahlster, W. (Eds.). (1989). *User models in dialogue systems.* Berlin: Springer.

Komatani, K., Ueno, S., Kawahara, T., & Okuno, H. G. (2005). User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction, 15*, 169–183.

Krüger, A., Baus, J., Heckmann, D., Kruppa, M., & Wasinger, R. (2007). Adaptive mobile guides. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 521–549). Berlin: Springer.

Krulwich, B. (1997). Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *AI Magazine, 18*(2), 37–45.

Langley, P., & Fehling, M. (1998). *The experimental study of adaptive user interfaces* (Tech. Rep. 98-3). Palo Alto, CA: Institute for the Study of Learning and Expertise.

Lanier, J. (1995). Agents of alienation. *interactions, 2*(3), 66–72.

Lesh, N., Rich, C., & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In J. Kay (Ed.), *UM99, User modeling: Proceedings of the Seventh International Conference* (pp. 23–32). Vienna: Springer Wien New York.

Li, X., & Ji, Q. (2005). Active affective state detection and user assistance with dynamic Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 35*(1), 93–105.

Linden, G., Hanks, S., & Lesh, N. (1997). Interactive assessment of user preference models: The automated travel assistant. In A. Jameson, C. Paris, & C. Tasso (Eds.), *UM97 User modeling: Proceedings of the Sixth International Conference* (pp. 67–78). Vienna: Springer Wien New York.

Linden, G., Smith, B., & York, J. (2003, February). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76–80.

Liscombe, J., Riccardi, G., & Hakkani-Tür, D. (2005). Using context to improve emotion detection in spoken dialog systems. *Proceedings of Eurospeech 2005, the Ninth European Conference on Speech Communication and Technology*, Barcelona, 1845–1848.

Lisetti, C. L., & Nasoz, F. (2004). Using noninvasive wearable computers to recognize human emotions from physiological signals. *EURASIP Journal on Applied Signal Processing, 11*, 1672–1687.

Litman, D. J., & Pan, S. (2002). Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction, 12*(2–3), 111–137.

Looi, C., McCalla, G., Bredeweg, B., & Breuker, J. (Eds.). (2005). *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology.* Amsterdam: IOI Press.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37*(7), 30–40.

McCarthy, K., Reilly, J., McGinty, L., & Smyth, B. (2005). Experiments in dynamic critiquing. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 175–182). New York: ACM.

McCreath, E., & Kay, J. (2003). I-ems: Helping users manage e-mail. In P. Brusilovsky, A. Corbett, & F. de Rosis (Eds.), *UM2003, User Modeling: Proceedings of the Ninth International Conference* (pp. 263–272). Berlin: Springer.

McCreath, E., Kay, J., & Crawford, E. (2006). I-ems—an approach that combines hand-crafted rules with learnt instance-based rules. *Australian Journal of Intelligent Information Processing Systems, 9*(1), 49–63.

McGrenere, J., Baecker, R. M., & Booth, K. S. (2002). An evaluation of a multiple interface design solution for bloated software. In L. Terveen, D. Wixon, E. Comstock, & A. Sasse (Eds.), *Human factors in computing systems: CHI 2002 conference proceedings* (pp. 164–170). New York: ACM.

Micarelli, A., Gasparetti, F., Sciarrone, F., & Gauch, S. (2007). Personalized search on the World Wide Web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 155–194). Berlin: Springer.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J., & Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM, 37*(7), 81–91.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education, 10*, 238–256.

Mitrovic, A., Suraweera, P., Martin, B., & Weerasinghe, A. (2004). DB-Suite: Experiences with three intelligent, web-based database tutors. *Journal of Interactive Learning Research, 15*(4), 409–432.

Mobasher, B. (2007). Data mining for web personalization. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 90–135). Berlin: Springer.

Müller, C., Wittig, F., & Baus, J. (2003). Exploiting speech for recognizing elderly users to respond to their special needs. *Proceedings of the Eighth European Conference on Speech Communication and Technology*, Geneva, pp. 1305–1308.

Norman, D. A. (1994). How might people interact with agents? *Communications of the ACM, 37*(7), 68–71.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review, 13*, 393–408.

Petrelli, D., & Not, E. (2005). User-centred design of flexible hypermedia for a mobile guide: Reflections on the HyperAudio experience. *User Modeling and User-Adapted Interaction, 15*(3/4), 303–338.

Picard, R. W. (1997). *Affective computing.* Cambridge, MA: MIT Press.

Rhodes, B. J. (2000). *Just-in-time information retrieval.* Unpublished doctoral dissertation, School of Architecture and Planning, Massachusetts Institute of Technology.

Rich, C., & Sidner, C. L. (1998). COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction, 8*, 315–350.

Rich, C., Sidner, C. L., & Lesh, N. (2001). COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine, 22*(4), 15–25.

Rich, C., Sidner, C., Lesh, N., Garland, A., Booth, S., & Chimani, M. (2005). DiamondHelp: A collaborative interface framework for networked home appliances. *5th International Workshop on Smart Appliances and Wearable Computing, IEEE International Conference on Distributed Computing Systems Workshops,* 514–519.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The adaptive web: Methods and strategies of web Personalization (pp. 291–234). Berlin: Springer.

Schiele, B., Starner, T., Rhodes, B., Clarkson, B., & Pentland, A. (2001). Situation aware computing with wearable computers. In W. Barfield & T. Caudell (Eds.), *Fundamentals of wearable computers and augmented reality* (pp. 511–538). Mahwah, NJ: Erlbaum.

Schneider, M., & Kiesler, S. (2005). Calling while driving: Effects of providing remote traffic context. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 561–570). New York: ACM.

Sears, A., & Shneiderman, B. (1994). Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction, 1*, 27–51.

Segal, R. B., & Kephart, J. O. (1999). MailCat: An intelligent assistant for organizing e-mail. *Proceedings of the Third International Conference on Autonomous Agents*, Seattle, 276–282.

Segal, R. B., & Kephart, J. O. (2000). Incremental learning in SwiftFile. In P. Langley (Ed.), *Machine Learning: Proceedings of the 2000 International Conference* (pp. 863–870). San Francisco: Morgan Kaufmann.

Shearin, S., & Lieberman, H. (2001). Intelligent profiling by example. In J. Lester (Ed.), *IUI 2001: International Conference on Intelligent User Interfaces* (pp. 145–151). New York: ACM.

Shneiderman, B., & Maes, P. (1997). Direct manipulation vs. interface agents. *interactions, 4*(6), 42–61.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., & Boydell, O. (2005). Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction, 14*(5), 383–423.

Soller, A. (2007). Adaptive support for distributed collaboration. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 573–595). Berlin: Springer.

Sumi, Y., & Mase, K. (2001). AgentSalon: Facilitating face-to-face knowledge exchange through conversations among personal agents. *Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, 393–400.

Sumi, Y., & Mase, K. (2002). Supporting the awareness of shared interests and experiences in communities. *International Journal of Human-Computer Studies, 56*(1), 127–146.

Surendran, A. C., Platt, J. C., & Renshaw, E. (2005). Automatic discovery of personal topics to organize e-mail. *Proceedings of the Second Conference on E-mail and Anti-Spam*, Stanford, CA.

Teevan, J., Dumais, S., & Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. *Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 449–456.

Terveen, L., & McDonald, D. W. (2005). Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction, 12*(3), 401–434.

Trewin, S. (2004). Automating accessibility: The dynamic keyboard. *Proceedings of ASSETS 2004,* 71–78.

Tsandilas, T., & schraefel, m. (2004). Usable adaptive hypermedia. *New Review of Hypermedia and Multimedia, 10*(1), 5–29.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education, 15*(3), 147–204.

Vivacqua, A., & Lieberman, H. (2000). Agents to assist in finding help. In T. Turner, G. Szwillus, M. Czerwinski, & F. Paternò (Eds.), *Human factors in computing systems: CHI 2000 conference proceedings* (pp. 65–72). New York: ACM.

Wainer, H. (Ed.). (2000). *Computerized adaptive testing: A primer.* Hillsdale, NJ: Erlbaum.

Wexelblat, A., & Maes, P. (1997). Issues for software agent UI. Unpublished manuscript, available from http://citeseer.ist.psu.edu/128549.html on 18 March 2007. Cited with permission.

Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance.* Upper Saddle River, NJ: Prentice Hall.

Wilde, G. J. (1998). Risk homeostasis theory: An overview. *Injury Prevention, 4*(2), 89–91.

Wolfman, S. A., Lau, T., Domingos, P., & Weld, D. S. (2001). Mixed initiative interfaces for learning tasks: SMARTedit talks back. In J. Lester (Ed.), *IUI 2001: International Conference on Intelligent User Interfaces* (pp. 167–174). New York: ACM.

Yacoub, S., Simske, S., Lin, X., & Burns, J. (2003). Recognition of emotions in interactive voice response systems. *Proceedings of Eurospeech 2003, the Eighth International Conference on Speech Communication and Technology*, Geneva, 729–732.

Zhou, M. X., & Aggarwal, V. (2004). An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. *Proceedings of UIST 2004*, 227–236.

Zhou, M. X., Wen, Z., & Aggarwal, V. (2005). A graph-matching approach to dynamic media allocation in intelligent multimedia interfaces. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 114–121). New York: ACM.

Ziegler, C., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. *Proceedings of the 2003 International World Wide Web Conference*, 22–32.

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction, 11*, 129–158.

·23·

MOBILE INTERACTION DESIGN IN THE AGE OF EXPERIENCE ECOSYSTEMS

Marco Susani
Advanced Concepts Group, Motorola Experience Design

The Mobile Experience **460**
Interacting With Networked Ecosystems 460
Content, Communication, Control 460
From Interaction to Experience 460
Spheres of Interaction **461**
Sphere 1: Of Humans and Devices 461
Cars and Motorcycles 461
The Ritual Power of Hands 461
Form Follows Gesture 462
The Sense of Touch 462
Product Species 463
Sphere 2: Mental Models (and Their Representation) 464
Mental Models 464
Sphere 3: Context 464
Physical Context 464
Smart Objects and Active Ambient 465
Sphere 4: Communities 465
Sphere 5: Content 466
Deconstructed Content 466
Narrowcast 466
Seamless Experience Ecosystems 467
Adapting experiences to evolving behaviors 467
Shaping an experience without controlling the full ecosystem 467
Embracing the idea of human-to-human (and human-to-content) mediation rather than human-machine interaction 467

THE MOBILE EXPERIENCE

Interacting With Networked Ecosystems

The parallel evolution of network technologies and social behaviors is transforming the character of interactive systems.

The conventional model of the individual working in front of a personal computer is challenged by innovative ways of accessing content and communication; networked mobile wireless devices grow their functionality, handheld interfaces become richer, mobile distributed delivery of content will be fully integrated with conventional media, and new social forms of communication are emerging.

The mobile component of this system of interactions, centered on the device formerly known as the cell phone," has been among the most impressive techno-cultural phenomena of the past 10 years, and its evolution is still changing drastically the way human interaction is conceived. Mobility is only one aspect of this revolution; mobile devices also happen to be much more related to the individuals that use them, and follow constantly both their private and work life. Thus, the interaction with these tools is characterized by a combination of functional and emotional aspects.

Present interaction models, even the ones developed respecting the rules of usability, often lack fundamental emotional qualities that are needed to support and increase existing social behaviors and rituals.

The paradigm of pervasive and ubiquitous networking, instead of gravitating around the personal computer, is taking the shape of an ecosystem of mobile and static platforms, for individual and collective use, all connected to the network, and their coordination is taking the shape of a choreography of interactions to provide access to integrated content and communication.

"Convergence" seems finally to take shape, centered on three platforms that dominate this experience ecosystem:

- Virtually all personal computers are on the net, and most of them can access the net via mobile wireless technologies.
- Two billion cell phones, most of them capable to network both voice and net, are in the hands of people of all cultures and statuses.
- The predictable transition of cable TV to IP networked TV is about to transform the paradigm of interacting with video, movies, and TV content. Around the TV, other distributed devices (the evolution of the "device formerly known as the remote control") will increase the mobility within the home.
- Embedded networking capabilities are growing in other "appliances," such as cars, or in any "thing" (via the RFID technology), and in active environments.
- All of the above is going to be connected with a multiplicity of technologies; the integration of wired and wireless, and the integration of different technologies within each one of these domains (e.g., cellular networks talking to local wireless networks at different scales, from Bluetooth to wifi to RFID), transforms the picture of a network into a flexible, capable, seamless, "opportunistic" infrastructure that can take advantage of any available connectivity.

Similar social and cultural disruptions are happening in parallel with this fully developed technology disruption:

- Format and genres of communication integrate and hybridize;
- Content becomes deconstructed and pervasive;
- Based on the intersection of the two aspects above, new types of communities are emerging, and new social models growing around the generation and distribution of content explore the space between one-to-one communication (the evolution of the telephone) and broadcast (the evolution of mass media).

Devices of different type become multiple points of access to the network, and the Net is much more embedded in the everyday reality, both because it is more integrated with spaces and artifacts of everyday life, and because it influences more and more human activities and social systems.

Content, Communication, Control

The original interaction with personal computers focused on doing (writing a document, make calculations on a spread sheet, and so on), and its physical context was a single individual user sitting in front of his computer, isolated from the surrounding physical space. Today, the merging of computer technologies with telecommunication technologies and wireless network access calls for a completely different paradigm: computers, mobile devices, and networks are not only a complex set of tools to "do" things, to perform tasks. Instead, they are above all gates to communication media that connect people, hybrid physical and digital collaborative spaces to meet others, and tools to access published, dynamic, and ubiquitous content.

The interfaces of these devices aren't anymore exclusively mediators between a machine and an individual, but much more often are the interfaces that structure the user's communication mental model, or interfaces that organize and categorize in an accessible way the vast space of networked content.

Simplicity, the key attribute that allowed in the past easy access to the functionality of a "machine," becomes today a much more sophisticated concept, because a "simple" access to the user's sphere of communication and content cannot come only from a functional, efficient organization of the interaction, but is much more rooted in subjective and cultural aspects, such as the mental construct of the user's social universe.

In this chapter we'll investigate the specifics of interaction with a multiplicity of mobile networked devices, and we'll see how the whole paradigm of networked reality changes the way we think about HCI. We'll look at any networked device, whether mono-functional or multi-functional, such as cell phones, networked photo-cameras, networked digital-music players, digital TV, and so on, as points of access to this vast ecosystem of networked content, communication, and control.

From Interaction to Experience

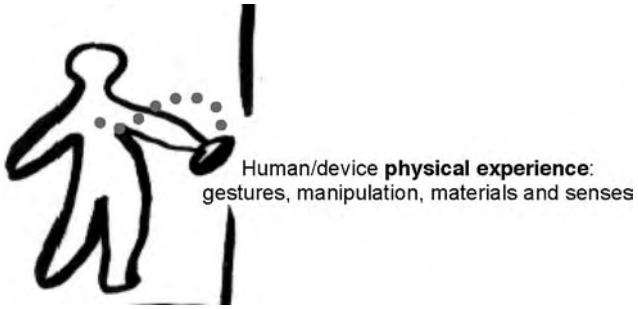
In a networked universe that has no center, we'll use the human user as a subjective viewer of the whole experience ecosystem.

Around this hypothetical user, we'll describe different concentric "spheres" of interaction. The first "sphere," closest to the user, corresponds to what usually is taken in consideration when designing a computer system: the physical interaction with a device. The other "spheres," growing progressively to include multiple elements of this experience ecosystem, will include topics related to human-to-human communication or access to content, or interacting with objects in a space.

When mobile devices are enriched by multiple functions and applications, this whole ecosystem is what actually influences the overall experience of the user. This is the new territory for designers to act and define the character of a product, the features of a service, or the identity of a medium.

SPHERES OF INTERACTION

Sphere 1: Of Humans and Devices



The first sphere of interaction, the most intimate connection between a human and a device, is the physical manipulation. The very nature of handheld devices makes this manipulation a key component of the relationship between user and system. Although handheld devices like mobile phones have conventional keypads and a screen that sometimes mimics the navigation in the GUI (Graphical User Interface) of the overall experience is extremely different from using a PC. The manipulation and relation with the hand, and in some respect with the rest of the body (such as through placing the device in a bag or in the pocket), roots these objects in the history of tools that are real extensions of the hand, such as pens. Also, the handheld nature of mobile devices highlights social aspects, such as the proxemic role of the object, like handing over a phone to show a picture to a friend, for example. In this context, rich and sophisticated socio-cultural aspects complement the basic ergonomic components of the interaction.

Cars and Motorcycles

If you talk to a passionate biker, you'll most often hear a comparison between the emotional aspects of driving a motorcycle compared to the "dullness" of driving a car; riding a bike is a richer sensorial experience that involves the whole body. While one drives a car by "codified" controls, proper interfaces that act on the machine, riding a bike involves an extension of the body. Actually, riding a bike is a hybrid between some coded controls (such as clutch and brakes) and "natural" proprioceptive controls (such as moving the body while making a turn).

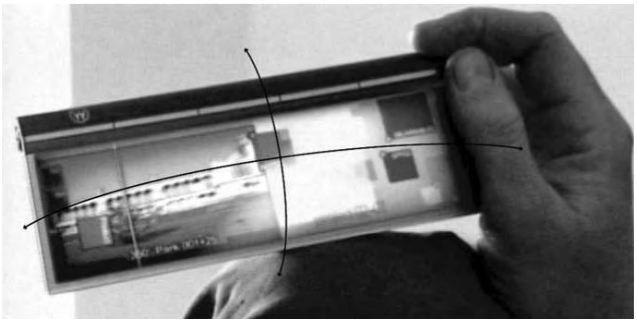
Consequently, a car may be perceived as "passive," while riding a bike is perceived as an active and engaging experience. Furthermore, riding a bike puts you in direct, immediate connection with the surrounding environment. From a certain point of view, the same can be said of personal computers and cell phones. Computers' only relationship with the body is the flat space of the desk, the movements of fingers on the keyboard and the hand on the mouse. We use computers while sitting alone at a desk, our bodies still, and our minds isolated from the surrounding environment. Mobile devices such as cell phones are handheld, and the relationship with the hands is a fundamental character of the interaction. A PC provides an experience that atrophies our body to minuscule movements of the fingers and the wrist, while handling a cell phone is an experience that may involve larger gestures and has a richer relationship with the body. We use mobile phones while walking, sitting, standing, and driving. We use them in isolation or in a crowd. The experience of communicating via a mobile phone is only partially immersive; the environmental noise and the distractions of the environment balance this immersion, and provide meaningful context to our activity. Cell phones have transformed the way we approach others and the power of the context enriches the communication; rather than asking, "How are you?" when we start a conversation, we ask, "*Where* are you?" as if the environment was an integral part of the conversation. The limited size of the device, which is in some respects a limit to the interaction because the screen is small and crowded, actually helps provide a further element of context; with such a small screen, we're forced to see what's around it, and the interaction is always peripheral, never immersive, and always hybridized by the "real" experience of the world around.

The Ritual Power of Hands

Consider the way people take pictures with a cell phone. Sometimes the camera phone is pointed toward the subject of the photo like a camera, and sometimes the phone is used in a very different way to take pictures that are much more spontaneous and transient. To be fair, the way camera phones are used had been anticipated by a phenomenon grown around very conventional film-based cameras. In the early 1990s, users of a camera called Lomo started to take very spontaneous pictures by not aiming in the viewfinder. The Lomo phenomenon transformed the genre of photography with a disruption that was purely "behavioral": a new gesture, such as handling the camera without bringing it near the eye, revolutionized photography and made the images much more "instantaneous" and genuine. This gave birth to a new photographic genre that was called Lomography. In Lomography, the gesture *is* the photo. The way people manipulate camera phones is very similar. In this case, the instantaneous photo genre is reinforced by the fact that photos are taken to be shared, not to be kept. While in origin photography was conceived as a way of "conserving" reality by documenting it, networked photos are transient, not perennial. And the informality of the gesture of "catching" them is a fun-

damental part of both the interaction and the narratives of this genre of photography. Designing a form that facilitated or denied this spontaneity would be for a designer a way of confirming, or denying, this behavior, and as a consequence a way of influencing the genre of photographs taken with such a device. In this case, the whole form and shape of the device, rather than an interface or a control, is the fundamental component of the interaction. Better, the shape of the device *is* the interface. While this is relatively new in the history of human computer interaction, it is not new in the history of objects.

Form Follows Gesture



In the history of material culture, a history that includes, for example, ancient cooking utensils and working tools, the relationship between objects and hands is fundamental, and ends up defining the ritual relationship between humans and their tools. We are facing a kind of "calligraphic" use of an object. The idea of calligraphy, etymologically a "beautiful way of drawing or writing," is very much related to the idea of dignity of the gesture.

In addition to being a fundamental component of the use of the device, the overall shape, proportions, and mechanics of the object are also establishing the "character" of the object, and with that they influence the social dynamics around the usage of the object. The way John Travolta pulls the antenna of his flip phone with his teeth in Pulp Fiction (in the topical moment when his friend just shot a guy by error in a car) is integral part of his character, and of the violence in that scene. How different is this from the way Anna Magnani "hugs" the handset in a cinema rendition of Jean Cocteau's theatrical piece "La Voix Humaine," a love monologue "designed" around the relationship between the woman—a desperate lover who has been abandoned by her partner—and the handset of her phone, the only thread left between her and her lover. Or consider how different is a large phone hanging from the holster of a worker from a tiny phone held constantly in the hands of a minute Japanese teenager.

The manipulation of handheld devices also has a social role: sharing a photo sometimes means also grabbing the digital camera or the phone and handing it to somebody to watch at the screen. Even if the screen is small, sharing it with a friend is a rich social act. Handheld devices are part of the proxemic relationship between humans, other humans, and the social space that surrounds them.

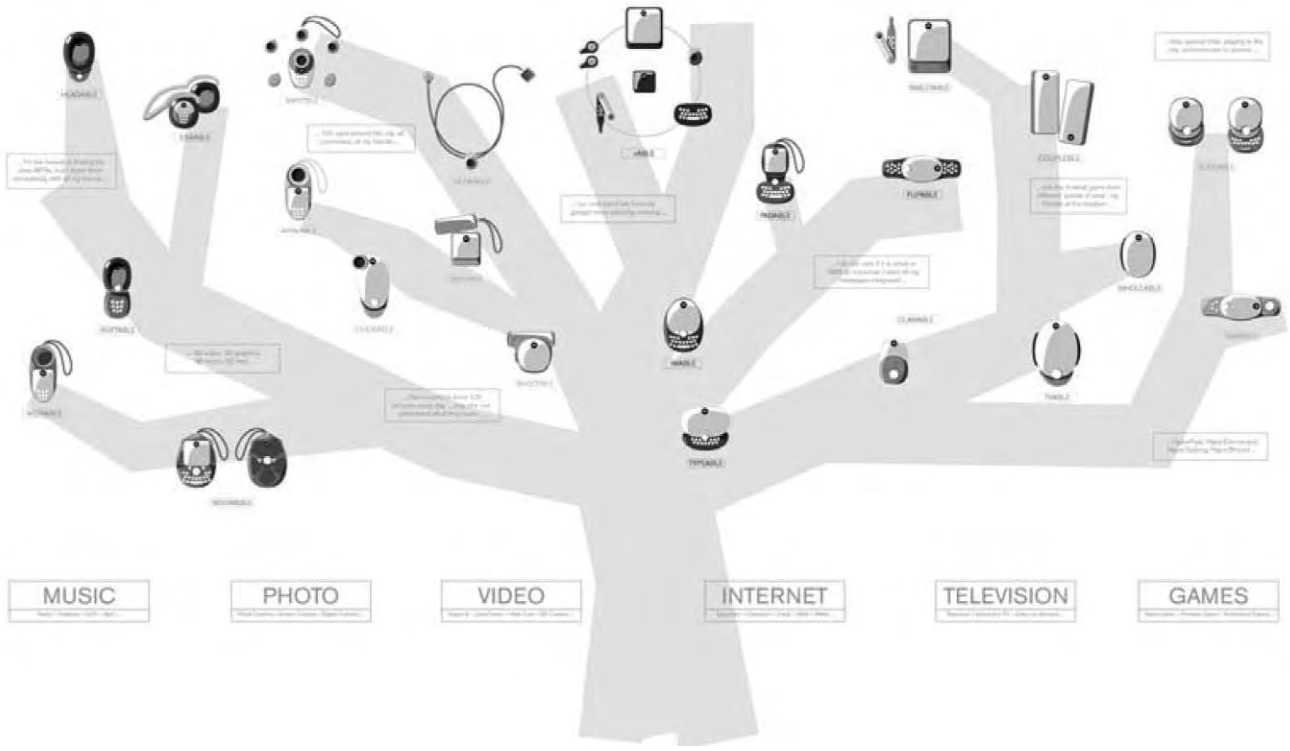
When devices such as cell phones are becoming so miniaturized that their shape is no more influenced by their technical components, their form would rather be defined by the way people manipulate them. And, as we described previously, the manipulation is only in part delimited by the mere rules of usability.

The Sense of Touch



The diffusion of keyboards and keypads have led to a kind of "buttonification" of the human being. This is definitely a reduction of the power of the sense of touch. Our senses are way more sophisticated than the basic on/off controls that they operate. In the case of handheld devices, this sensorial atrophy is more evident for at least two reasons. The first one is that while we hold the device with the full hand, five fingers plus the palm, we only use one finger to actually operate the keypad and receive tactile feedback. The second one is that, for the "peripheral" nature of the interaction with handhelds that we mentioned before, our interaction with a keypad ignores other fundamental spatial and proprioceptual clues, such as the horizon, our wrist position, and gravity. Some recent advances in sensing and feedback technologies have finally fought back this kind of sensorial deprivation that keypads have provoked. With the introduction of haptics, the primitive vibrators used in originally only to replace the ringer tone of phones are used to provide a broader tactile feedback. Haptic actuators amplified by the resonance of the whole mass of the handheld can stimulate the whole palm of the hand, or they could be used in a denser "resolution" to provide localized feedback and a more meaningful haptic "message" to parts of the hand. Similarly, the introduction of accelerometers and gravity sensors allows the device to be "conscious" of its absolute position in space and its relative position to the hand of the user. This allows users to take full advantage of the proprioceptual character of handheld manipulation, and provides a fuller tactile stimulation as a feedback to this manipulation. When combined with a proper reflection of the proprioceptual character of the GUI, the overall combination of enriched tactile and visual contextualized experience finally takes full advantage of the handheld nature of mobile devices. Localized buttons will remain for a long time, but combined with these additional sensory messages they will provide a much richer sensorial experience.

Product Species



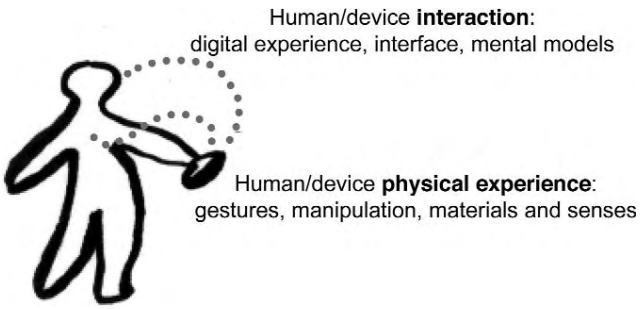
Even when the interaction with devices is properly designed and based on the principles of usability, sometimes there seems to be an additional difficulty in understanding the functions of an object, a subtler barrier to its adoption. An object may fail to deliver a clear message of its functionality to the user. The issue in this case is not "how easy is it to perform a task with this device?" but "what does this device do?" or, even more, "what is this?" In the past, a good rule to introduce a new object has been to leverage an analogy with a known, familiar object.

The reference to familiar devices and interactions could still apply today, but can also become too big of a limitation, a restriction of new functionalities or richer paradigms that simply do not have any parallel with anything familiar. Too many references are challenged: networked TV has no channels; mobile phones and house phones hybridize; e-mail and IM integrate; camera, telephone, and video functionalities converge in the same device. Communication paradigms such as blogging, or content distribution models such as podcasting, simply do not have any reference to past, established, familiar models.

Inventing new object archetypes, as much as it is an oxymoron, may be the only way out of this impasse—developing new product types and new interaction paradigms when playing analogy to existing archetypes is just not applicable. The same applies to mono-functional devices in comparison to multifunctional ones; the history of interaction design is full of successes of simple, mono-functional devices that leveraged this analogy. From the Palm PDA to the iPod, all fall in the category of devices that "do one thing and do it right." In the future, however, reducing the functionality in order to achieve simplicity of use may be no longer possible. Multifunctionality may offer too much of an advantage to be compromised, and new generations of users, younger hyper-taskers, wouldn't necessarily appreciate oversimplification.

So, the challenge for a designer is to ensure that an object would be "understood" while preserving multifunctionality and richness of experience. Designers would on the one side develop a richer perspective on visual affordances, one that could accommodate multiple and partially ambiguous affordances related to different usages. On the other side, beyond affordances, designers need to develop a better understanding of what constitutes a recognizable, understandable "species" of objects. That is to say, rather than seeking the familiarity with an known object or function "as-is," developing a more sophisticated understanding of how familiarity(ies), or just components of this, can be carried over along the evolution of an object, or carried across a discontinuity in the evolution of such an object or functionality. The result will be the transformation of the simple concept of affordance into a richer notion of "objecthood," defined as the ability of humans to recognize and perceive correctly the world of objects, and the mutual ability of objects to be understandable and to overtly disclose their functionality. That is, for both humans and objects to develop their relationship around a sixth sense, the "sense of meaning."

Sphere 2: Mental Models (and Their Representation)



Mental Models

In personal computers, the metaphor of the "desktop" and the "point and click" in that digital space starts to feel the "competition" of other models of interaction, namely the information architectures organized around e-mail and IM, that today are often dominant compared to the desktop access to files and folders.

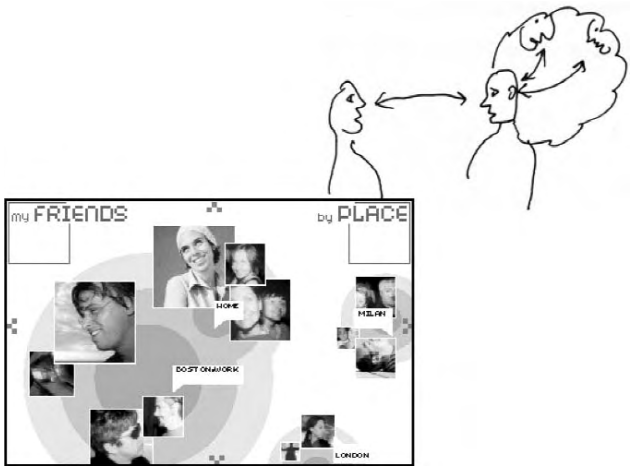
The mental construct of the interaction with mobile wireless media is intrinsically centered on communication; the phonebook, and its extension to include presence, status, and, in perspective, location, is often the master mental model that represents our communication universe. The phonebook acts as a point of departure for a broad series of actions: starting a call, generating a message, initiating a textual chat, checking the online status of our "buddies," or sending a picture to somebody. However, with the diffusion of other ways of contextualizing the interaction, such as location detection, a user may decide to switch to a different model and, for example, "view by location" and reorganize the communication, mapping it over the actual physical surroundings. In this model, a user may reorganize the phonebook to check first "who's around me" in that moment. Even if the idea of communicating in physical proximity may seem paradoxical, cell phones are often used to create an additional communication channel (texting or sharing images, for example) with people who are not remote, or to manage meetings with people in physical proximity (e.g., a group of friends in the same city arranging a dinner together).

Other mental models use incoming events as an organizational context; an incoming call may trigger a reorganization of the interface around the person who calls (bringing in the foreground previous messages or pictures exchanged with that person); or walking in front of a place may trigger some location-based content and contextual actions around that content; or an approaching event, such as a meeting, can trigger other information such as the names of the participants, their status, and their location.

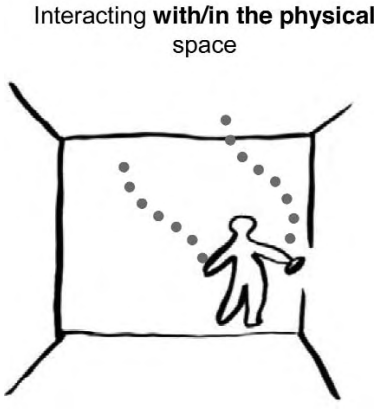
While the taxonomies arranged around the sphere of communication may work better for phone-first devices, the proliferation of functions (imaging, video, music management, e-mail, etc.) could suggest taxonomies based around a clear distinction of applications, or around categories of content (my photos, my music, my phonebook, etc.).

The dominance of the former model (centered on communication) or the latter (centered on content or applications) is not yet clear. For a long time we'll probably continue to have hybrids with the co-presence of multiple categorizations.

What is common to both these models and different from the old desktop metaphor is that they have a less direct analogy with an interaction space—such as the physical space of the desktop—and they have more reference to mental constructs. As such, these models could be referred to as "cosmographies." While a metaphor like the desktop implies having a reference with something that exists, a cosmography is a "general description of the world or universe," and this could embrace descriptions of both the material world and of abstract concepts—a mental model of something that eventually does not have any relation with the material reality.



Sphere 3: Context

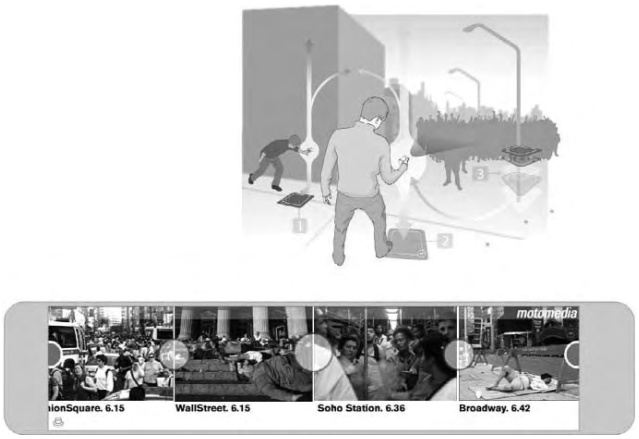


Physical Context

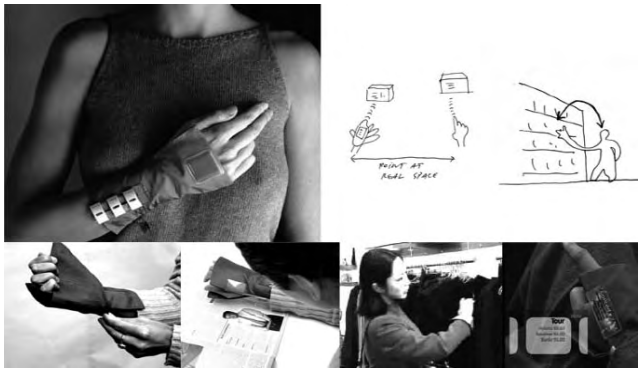
Mobile interaction cannot be isolated from its context. While the interaction between a user and a personal computer tends to be immersive and central, and is agnostic of the physical context of the user, the interaction with mobile devices is typically contextual and peripheral.

Primitive graphic menus and icons arranged in the small screen, a kind of translation of the conventional PC, are the dominant GUI paradigm in wireless mobile media, but they are only the most superficial aspect of the interaction with devices like cell phones. Multiple other aspects of interaction with cell phones are less tangible, yet more important, than the keypad-and-screen UI. The whole experience ecosystem of the interaction with wireless media includes the combination of the interaction with the physical device, the interaction with its services, and with the physical environment.

Context is what makes mobile interaction much richer than computer interaction. Mobility means that to be *more connected* to the physical place, not *more disconnected* from it; communication relates to *where* we are and *who* is around us. Location-based services, mobile imaging, and mobile blogging extend this concept to content itself: content accessed and generated from a mobile platform has the potential to be heavily contextualized over the physical environment.



Smart Objects and Active Ambient



Smart tags are very tiny, low-cost, wireless microcomputers. They only cost a few cents and they can be attached to anything—items of clothing, food cans, spare parts for cars, electric irons, and so on. The tags can be programmed to provide information about the items to which they are attached. They are able to identify a product and network this information, and can also take note of their environment, recording changes in temperature and so on. Their broad adoption is changing the way logistics are organized around raw materials and products traveling to transformation and to markets.

Smart tags also enable a minute level of interaction with the physical space. Cell phones equipped with RFID tags are already enabling financial transactions such as paying a ticket on the underground or replacing a credit card. The extension of this level of interaction may support the paradigm of "point and click at real environments." Handheld devices could be used to trigger contextual information when pointed at other objects or parts of the environment.

Finally, the connection of these tags to an integrated information system able to tell, at any moment, about of the life of any item of produce, for example, may allow the quality, the origin, or the value, to be tracked in real-time. Produce that is fresh or older, in-season or off-season, near or far from its origin, with or without a track record of its growth, will have a different value. In this scenario, smart tags may enable the creation of *digital word-of-mouth*, similar to the spreading of information and knowledge by human word-of-mouth, connected to the system of goods. Attached to items will be information and knowledge. It is possible to describe this as a *knowledge aura*, and this is the most interesting aspect of future material-knowledge systems. Experiments on connecting *collective wisdom* with digital systems *on the field* have been undertaken in India since the 1990s. The fully fledged scenario will allow replication in a digital environment, of the wisdom of word-of-mouth know-how.

The whole idea of using advertisement before the purchase, and separate from the purchase experience, could be revolutionized by the presence of an aura of information that follows the product until the point of purchase. Through this aura of digital information, the product would "advertise itself" from the shelves of a shop. Even more intriguing is the idea that deeper information about a product, connected to similar experiences of purchase and use—what is known as a recommendation on a website—will be readily available at any moment that a product is encountered. The lure and temptations that attract a buyer will come directly from the product, from the background voices of people that bought and used it, and by the life history of the product—a kind of very detailed, dynamic, digital certificate of authenticity.

Sphere 4: Communities



Networked media are characterized by the hybridization of a multiplicity of communication genres. While 20 years ago the evolution of telephone technologies seemed to go toward hyper-realism—technologies focused on replicating, at best, face-to-face communication—the actual evolution of telecommunication

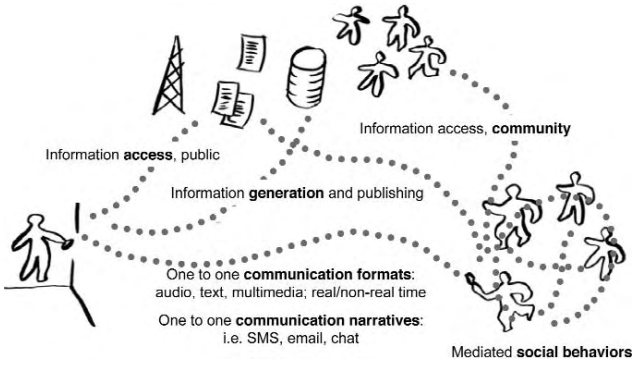
has denied this trend and pointed toward mediating communication in a form *different* from face-to-face. Text messaging via SMS created communication channel that was simply impossible to achieve with face-to-face—often more intimate, more enigmatic, than face-to-face, and complementary to that. Instant messaging has added the management of multiple one-to-one communications, also not possible in real face-to-face, and, later, blogging created a hybrid between one-to-one expression and broadcast. Finally, the multimedia capabilities of devices like camera phones allow the integration between communication and content sharing, making it possible to share experiences in an almost permanent way.

Mobile wireless media also are the ideal platform to support tribal social structures; mobile phones have favored the birth of tribal forms of few-to-few and few-to-many communication, and they revitalize the scale of the small circle of friends, of groups of people who are in constant touch, exchanging information and passions, and sharing images and music.

Different from both an individual that communicates one-to-one with his interlocutor and from a broadcaster with a passive mass audience, tribal models of communication define a dense flow of exchanges within a restricted circle of friends. The space for tribal relations, superimposition of the physical space and of the "digital territory of belonging," also amplifies information and messages while building a shared memory. The permanence of content in a shared space creates not a simple exchange of messages but the sedimentation, the memory, and the *construction* of a shared permanent experience.

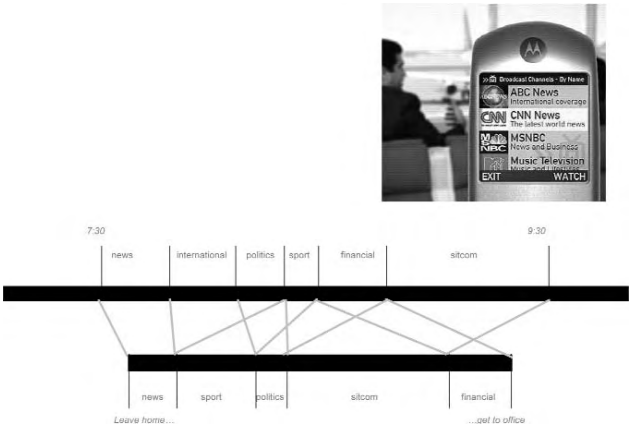


Sphere 5: Content



"Convergence" between media, for long just a buzzword, is happening for real, and this means a much richer interaction for the user. Richer communication—messaging, voice, and so on—is more and more integrated with content access, such as sharing music, videos, and images.

Deconstructed Content

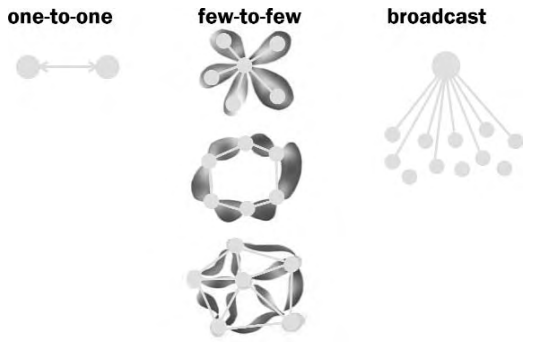


Mobile imaging is an already-established form of hybridization between communication and content capture. Camera phones as an imaging platform are better integrated with both the PC and the TV. Video content will soon follow, and this, combined with the integration of web content with TV productions, will create yet another "triple point" (mobile, PC, and TV and home systems) interaction with content. As soon as digital music players become networked themselves and mobile phones download music over-the-air, the music-content ecosystem will have continuity between environments and platforms.

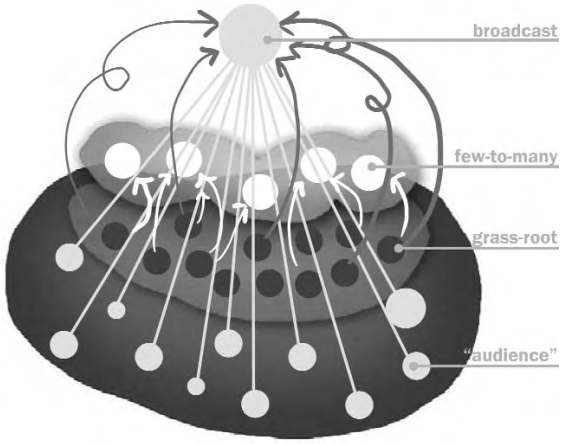
The first consequence this will have from an interaction point of view will be the separation of content creation from content delivery and the consequent "deconstruction" of content. The possibility of capturing images with different devices (camera phones or digital cameras) and accessing them in different modalities (in the device itself, shared with others or published on the web, or printed or accessed from a PC or a TV) are already evident signs of separation between creation and access, and examples of truly networked access to content. Mobile TV will not be an exact replica of TV delivered to home. The latter has already been transformed by Digital Video Recorders, and broadcasts have been separated from real time. Mobile TV will be even more disconnected from the real time of broadcast—often buffered and stored locally—but also re-associated with the "relative" time of the user; the possibility of caching content on a mobile device, combined with the need for "snacking" shorter versions of the same content during leftover time while commuting, for example, will make the idea of a personal, adapted palimpsest more realistic. Content will thus be deconstructed in different multiple formats (filling up many lengths, between the short trailer and the full-length movie) and consumed from different platforms.

Narrowcast

Podcasting and blogging are the signal of a larger trend that opens the possibility for everybody to become a publisher. In both cases the broadcast paradigm evolves from one-to-many (where the one is an author, an institutional publisher such as a TV station) to any-to-many (where anybody, with a limited in-



vestment and a much leaner technology, can become a publisher). But there is more. Both blogging and podcasting are creating communities of adepts that do not behave like a conventional passive audience of broadcast mass media. They feed back and create a closer circle of communication that constitutes an actual "narrowcast" environment, a scale of publishing unheard of before. There is no solution of continuity between the one-to-one paradigm of telephony that extends to the larger scale of a tribe of friends and the blogging and podcasting narrowcast that takes broadcast to a smaller scale. Also. communication and content are so hybridized that they become indistinguishable.



Seamless Experience Ecosystems

Today no interaction can be isolated in the confined space of human computer interaction; any consideration on the interaction between a user and a mobile device cannot ignore the complexity of the relational "ecosystems" that exist around this interaction.

The components of this complex networked experience are

1. A social ecosystem, as mobile devices go beyond the one-to-one communication to mediate very rich and complex system of social relationships;
2. A content ecosystem, since mobile devices allow access to images, music, and videos; and peer-to-peer content generation and sharing introduce a completely new perspective of content authoring;
3. A spatial ecosystem, because location-based services and communication strongly "ground" the interaction via mobile devices in the physical space;
4. A business ecosystem, because no single company or designer can control the overall experience, but can only shape the interaction from a partial point of control; and last, but not least
5. A platform ecosystem, since mobile interaction is integrated with interaction with other platforms, such as personal computers and digital TV.

The complexity of these ecosystems puts additional challenges to designers that want to shape the interaction.

Among these challenges are the following:

Adapting experiences to evolving behaviors. Convergence will change drastically the way users perceive their sphere of content and communication, and sometimes will give up familiar mental models to embrace new paradigms may actually facilitate the ease of use. This is in apparent contrast with one of the rules of user-centered design, which is to refer to familiar models and avoid disruptions. For example, the potential elimination of "channels" when TV content moves over IP may help embrace a new paradigm that has no reference to the old "zapping." In the same way, embracing the paradigm of one-phone-number-per-person may confuse users at first, but will actually facilitate the transition to a communication paradigm where the user, not the device, is the one to be reached. A well-crafted, user-oriented approach may distinguish between the respect of familiarities that are structurally embedded in human perception and pushing back resistance of old mental habits that may actually hinder innovation and cause disruptions.

Shaping an experience without controlling the full ecosystem. The design culture has long insisted on the protection of the brand and on establishing a "signature" experience to a device or a service. In the future it will be very rare, however, that a single brand, thus a single designer, would control the full experience ecosystem. Two approaches to this are possible: one is a resistance to sharing components of the experience with other brands; another is embracing the complexity of the system and finding lean, smart ways to shape the experience without actually owning and controlling all the elements. The latter seems to be a more open-minded, relaxed way to accept changes in the environment a designer operates, while the former may end up in a stubborn, desperate attempt to protect a territory that simply cannot be protected anymore. In other words, it is better to adapt and shape an experience by seeding signature elements in few key areas rather than working on a more rigid model and risking losing all the control.

Embracing the idea of human-to-human (and human-to-content) mediation rather than human-machine interaction. The HCI discipline is rooted in the interaction between a human and a machine (or a system) and focused on the shift from a mechanical interaction to a logical one. While this competence remains true for the broad majority of devices we are designing today, it is also true that another aspect, the one of designing interfaces to access other humans and content

via a mediating interface, is becoming more and more dominant. A deeper understanding of human patterns of communication, and of the interactions between interfaces of devices and genres of communication, is needed. In the same way, the understanding of how innovative ways of interacting with content may shape new types and formats of content is also needed. Media studies were born in a condition in which the media genre (e.g., the TV programs), the transport technology (via aerial or cable TV), the device of access (the TV set), the method of interaction (zapping through channels), the method of delivery (broadcast) and the interaction environment (a couch potato in a living room) of a given medium were linked and homogeneous. Today, a certain type of content (e.g., a TV program) may be delivered in a multiplicity of formats (real time or cached, in snippets or in a linear fashion) to extremely different platforms and environments (at home or while mobile, to a passive individual or annotated and redistributed within a community). This poses new challenges to designers than need to shape at the same moment an innovative interaction and an innovative medium genre.

Experience ecosystems are reshaping the way human knowledge is grown and diffused. Interaction designers have the responsibility of both shaping a pleasurable, meaningful experience and of facilitating the chain of innovations that flows through new behaviors, new business models, and new technologies. Designers need to be the first to drive innovation in their own disciplines.

Images from interaction design projects developed by the Advanced Concepts Group in Motorola between 2000 and 2005.

•24•

TANGIBLE USER INTERFACES

Hiroshi Ishii
MIT Media Laboratory

Introduction **470**
From GUI to TUI **470**
Urp: An Example of TUI **470**
Basic Model of Tangible User Interface **472**
GUI **472**
TUI **472**
Tangible Representation as Control 472
Intangible Representation 473
Key Properties of TUI 473
Computational coupling of tangible representations to underlying digital information and computation 473
Embodiment of mechanisms for interactive control with tangible representations 473
Perceptual coupling of tangible representations to dynamic intangible representations 474
Genres of TUI Applications 474
Tangible Telepresence 474
Tangibles with Kinetic Memory 474
Constructive Assembly 474
Tokens and Constraints 474
Interactive Surfaces—Tabletop TUI 475
Continuous Plastic TUI 475
Augmented Everyday Objects 475
Ambient Media 475
TUI Instances 476
InTouch: Tangible TelePresence Through Distributed Synchronized Physical Objects 476
Curlybot: A Toy to Record and Play 476
Topobo: 3D Constructive Assembly With Kinetic Memory 477
mediaBlocks: Token and Constraint Approach 477
Digital Desk: Pioneer of Tabletop TUI 478
Sensetable and AudioPad: Tabletop TUI for Real-Time Music Performance 478
IP Network Design Workbench: Event Driven Simulation on Sensetable 479
Actuated Workbench: Closing a Loop of Computational Actuation and Sensing 480
SandScape: Continuous TUI for Landscape Design 480
musicBottles: Transparent Interface Based on Augmented Glass Bottles 481
Pinwheels: Ambient Interface to Information 482
Contributions of TUIs 483
Double Interactions Loop: Immediate Tactile Feedback 483
Persistency of Tangibles 483
Coincidence of Input and Output Spaces 483
Special Purpose vs. General Purpose 484
Space-Multiplexed Input 484
Conclusion **484**
Acknowledgments **485**
References **485**

INTRODUCTION

Where the sea meets the land, life has blossomed into a myriad of unique forms in the turbulence of water, sand, and wind. At another seashore, between the land of atoms and the sea of bits, we are now facing the challenge of reconciling our dual citizenships in the physical and digital worlds. Our visual and auditory sense organs are steeped in the sea of digital information, but our bodies remain imprisoned in the physical world. Windows to the digital world are confined to flat square screens and pixels, or "painted bits." Unfortunately, one can not feel and confirm the virtual existence of this digital information through one's hands and body.

Imagine an iceberg, a floating mass of ice in the ocean. That is the metaphor of Tangible User Interfaces. A Tangible User Interface gives physical form to digital information and computation, salvaging the bits from the bottom of the water, setting them afloat, and making them directly manipulatable by human hands.

FROM GUI TO TUI

People have developed sophisticated skills for sensing and manipulating their physical environments. However, most of these skills are not employed in interaction with the digital world today. A Tangible User Interface (TUI) is built upon those skills and situates the physically embodied digital information in a physical space. Its design challenge is a seamless extension of the physical affordance of the objects into digital domain (Ishii & Ullmer, 1997, 2000).

Interactions with digital information are now largely confined to Graphical User Interfaces (GUIs). We are surrounded by a variety of ubiquitous GUI devices such as personal computers, handheld computers, and cellular phones. The Graphical User Interface (GUI) has been in existence since the 1970s and the first appeared commercially in the Xerox 8010 Star System in 1981 (Smith, 1982). With the commercial success of the Apple Macintosh and Microsoft Windows, the GUI has become the standard paradigm for Human Computer Interaction (HCI) today.

GUIs represent information (bits) with pixels on a bitmapped display. Those graphical representations can be manipulated with generic remote controllers such as mice and keyboards. By decoupling representation (pixels) from control (input devices) in this way, GUIs provide the malleability to emulate a variety of media graphically. By utilizing graphical representation and "see, point, and click" interaction, the GUI made a significant improvement over its predecessor, the CUI (Command User Interface) which required the user to "remember and type" characters.

However, interactions with pixels on these GUI screens are inconsistent with our interactions with the rest of the physical environment within which we live. The GUI, tied down as it is to the screen, windows, mouse, and keyboard, is utterly divorced from the way interaction takes place in the physical world. When we interact with the GUI world, we cannot take advantage of our dexterity or utilize our skills for manipulating various physical objects, such as building blocks, or our ability to shape models out of clay.

Tangible User Interfaces (TUIs) aim to take advantage of these haptic interaction skills, which is a significantly different approach from GUI. The key idea of TUIs is to give physical forms to digital information. The physical forms serve as both representations and controls for their digital counterparts. TUI makes digital information directly manipulatable with our hands, and perceptible through our peripheral senses, by physically embodying it.

Tangible User Interface serves as a special-purpose interface for a specific application using explicit physical forms, while GUI serves as a general-purpose interface by emulating various tools using pixels on a screen.

TUI is an alternative to the current GUI paradigm, demonstrating a new way to materialize Mark Weiser's (1991) vision of Ubiquitous Computing of weaving digital technology into the fabric of a physical environment and make it invisible. Instead of making pixels melt into an assortment of different interfaces, TUI uses tangible physical forms that can fit seamlessly into a users' physical environment.

This chapter introduces the basic concept of TUI in comparison with GUI, early prototypes of TUI that highlight the basic design principles, and design challenges that TUI needs to overcome.

URP: AN EXAMPLE OF TUI

To illustrate basic TUI concepts, we introduce "Urp" (Urban Planning Workbench) as an example of TUI (Underkoffler & Ishii, 1999). Urp uses scaled physical models of architectural buildings to configure and control an underlying urban simulation of shadow, light reflection, wind flow, and so on. (Photo 24.1). In addition to a set of building models, Urp also provides a variety of interactive tools for querying and controlling the parameters of the urban simulation. These tools include a clock tool to change the position of sun, a material wand to change the building surface between bricks and glass (with light reflection), a wind tool to change the wind direction, and an anemometer to measure wind speed.

The physical building models in Urp cast digital shadows onto the workbench surface (via video projection), corresponding to solar shadows at a particular time of day. The time of day, representing the position of the sun, can be controlled by turning the physical hands of a "clock tool" (Photo 24.2). The building models can be moved and rotated, with the angle of their corresponding shadows transforming according to their position and the time of day.

Correspondingly, moving the hands of the clock tool can cause Urp to simulate a day of shadow movement between the situated buildings. Urban planners can identify and isolate inter-shadowing problems (shadows cast on adjacent buildings), and reposition buildings to avoid areas that are needlessly dark,or maximize light between buildings.

A "material wand" alters the material surface properties of a building model. By touching the material wand to a building

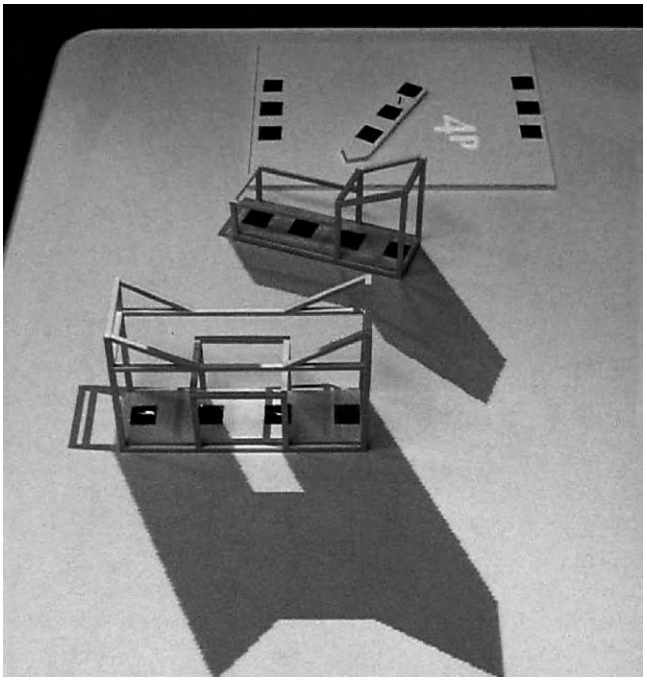


PHOTO 24.1. Urp and shadow stimulation.

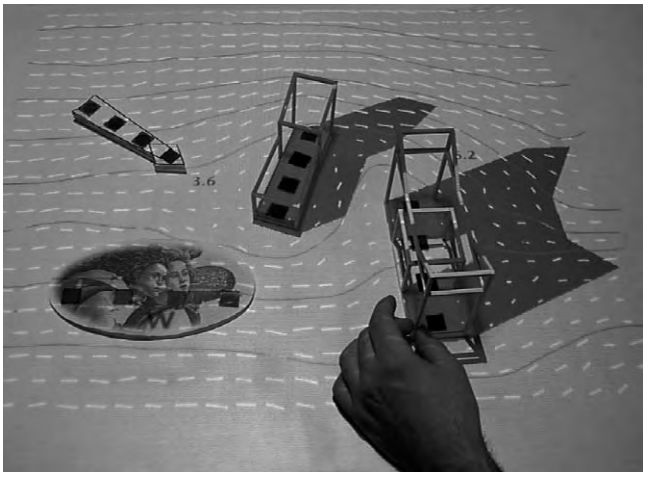


PHOTO 24.2. Urp and wind stimulation.

model, the building surface material is switched from bricks to glass, and a projected reflection of sunlight appears to bounce off the walls of the building. Moving the building allows urban designers to be aware of the relationship between the building reflection and other infrastructure. For example, the reflection off the building at sundown might result in distraction to drivers on a nearby highway. The designer can then experiment with altering the angles of the building to oncoming traffic or moving the building further away from the roadway. Tapping again with the material wand changes the material back to brick, and the sunlight reflection disappears, leaving only the projected shadow.



PHOTO 24.3. inTouch.

By placing the "wind tool" on the workbench surface, a wind flow simulation is activated based on a computational fluid dynamics simulation, with field lines graphically flowing around the buildings. Changing the wind tool's physical orientation correspondingly alters the orientation of the computationally simulated wind. Urban planners can identify any potential wind problems, such as areas of high pressure that may result in had-to-open doors or unpleasant walking environments. An "anemometer" object allows point monitoring of the wind speed (Photo 24.3). By placing the anemometer onto the workspace, the wind speed of that point is shown. After a few seconds, the point moves along the flow lines, to show the wind speed along that particular flow line. The interaction between the buildings and their environment allows urban planners to visualize and discuss inter-shadowing, wind, and placement problems.

In Urp, physical models of buildings are used as tangible representations of digital models of the buildings. To change the location and orientation of buildings, users simply grab and move the physical model as opposed to pointing and dragging a graphical representation on a screen with a mouse. The physical forms of Urp's building models, and the information associated with their position and orientation upon the workbench, represent and control the state of the urban simulation.

Although standard interface devices for GUIs, such as keyboards, mice, and screens, are also physical in form, the role of the physical representation in TUI provides an important distinction. The physical embodiment of the buildings to represent the computation involving building dimensions and location allows a tight coupling of control of the object and manipulation of its parameters in the underlying digital simulation.

In Urp, the building models and interactive tools are both physical representations of digital information (shadow dimensions and wind speed) and computational functions (shadow interplay). The physical artifacts also serve as controls of the underlying computational simulation (specifying the locations of objects). The specific physical embodiment allows a dual use in representing the digital model and allowing control of the digital

representation. In the next section, the model of TUI is introduced in comparison with GUI to illustrate this mechanism.

BASIC MODEL OF TANGIBLE USER INTERFACE

The interface between people and digital information requires two key components: input and output, or control and representation. *Controls* enable users to manipulate the information, while *external representations* are perceived with the human senses. Figure 24.1 illustrates this simple model of a user interface consisting of control, representation, and information.

In the Smalltalk-80 programming language (Burbeck, 1992; Goldberg, 1984), the relationship between these components is illustrated by the "model-view-controller" or "MVC" archetype, which has become a basic interaction model for GUIs.

Drawing from the MVC approach, we have developed an interaction model for both GUI and TUI. We carry over the "control" element from MVC, while dividing the "view" element into two subcomponents: tangible and intangible representations, and renaming "model" as "digital information" to generalize this framework to illustrate the difference between GUI and TUI.

In computer science, the term *representation* often relates to the programs and data structures serving as the computer's internal representation (or model) of information. In this article, the meaning of "representation" centers upon external representations—the external manifestations of information in fashions directly perceivable by the human senses that include visual, hearing, and tactile senses.

GUI

In 1981, the Xerox Star workstation set the stage for the first generation of GUI (Johnson et al., 1989; Smith, 1982), establishing the "desktop metaphor" which simulates a desktop on a bitmapped screen. The Star workstation was the first commercial system that demonstrated the power of a mouse, windows, icons, property sheets, and modeless interaction. The Star also set several important HCI design principles, such as "seeing and pointing vs. remembering and typing," and "what you see is what you get (WYSIWYG)." The Apple Macintosh brought this new style of HCI to the public's attention in 1984, creating a new trend in the personal computer industry. Now, the GUI is widespread, largely through the pervasiveness of Microsoft Windows, PDAs, and cellular phones.

GUI uses windows, icons, and menus made of pixels on bitmapped displays to visualize information. This is an intangible representation. GUI pixels are made interactive through general "remote controllers" such as mice, tablets, or keyboards. In the pursuit of generality, GUI introduced a deep separation between the digital (intangible) representation provided by the bitmapped display, and the controls provided by the mouse and keyboard.

Figure 24.2 illustrates the current GUI paradigm in which generic input devices allow users to remotely interact with digital information. Using the metaphor of seashore that separates a sea of bits from the land of atoms, the digital information is illustrated at the bottom of the water, and mouse and screen are above sea level in the physical domain. Users interact with the remote control, and ultimately experience an intangible external representation of digital information (display pixels and sound).

TUI

Tangible User Interface (TUI) aims at a different direction from GUI by using tangible representations of information that also serve as the direct control mechanism of the digital information. By representing information in both tangible and intangible forms, users can more directly control the underlying digital representation using their hands.

Tangible Representation as Control

Figure 24.3 illustrates this key idea of TUI to give tangible (physical and graspable) external representation to the digital information. The tangible representation helps bridge the boundary between the physical and physical worlds. Also notice that the tangible representation is computationally coupled to the control to the underlying digital information and computational models. Urp illustrates examples of such couplings, including

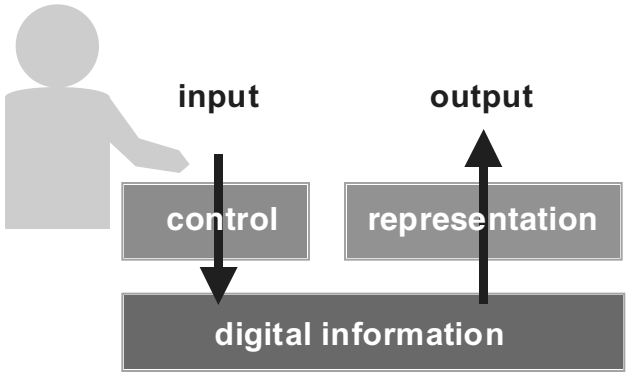


FIGURE 24.1. User interface model.

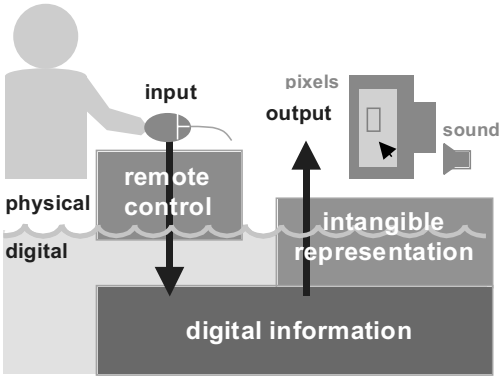


FIGURE 24.2. GUI model.

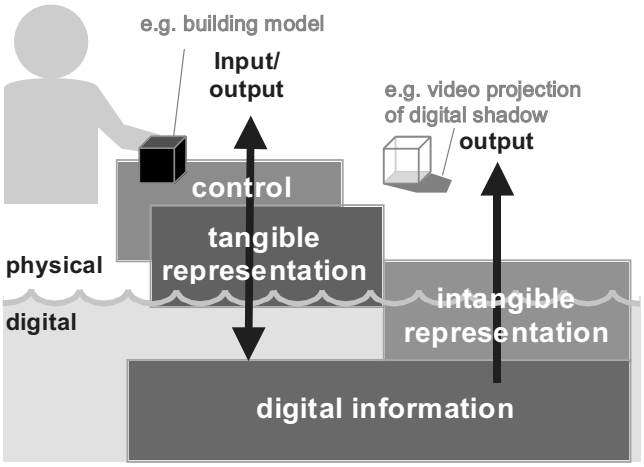


FIGURE 24.3. TUI model.

the binding of graphical geometries (digital data) to the physical building models, and computational simulations (operations) to the physical wind tool. Instead of using a GUI mouse to change the location and angle graphical representation of a building model by pointing, selecting handles, and keying in control parameters, an Urp user can grab and move the building model to change both location and angle.

The tangible representation functions as an interactive physical control. TUI attempts to embody the digital information in physical form, maximizing the directness of information by coupling manipulation to the underlying computation. Through physically manipulating the tangible representations, the digital representation is altered. In Urp, changing the position and orientation of the building models influences the shadow simulation, and the orientation of the "wind tool" adjusts the simulated wind direction.

Intangible Representation

Although the tangible representation allows the physical embodiment to be directly coupled to digital information, it has limited ability to represent or change many material or physical properties. Unlike malleable pixels on the computer screen, it is very hard to change a physical object in its form, position, or properties (e.g., color or size) in real time. In comparison with malleable "bits," "atoms" are extremely rigid, taking up mass and space.

To complement this limitation of rigid "atoms," TUI also utilizes malleable representations, such as video projections and sounds, to accompany the tangible representations in the same space to give dynamic expression of the underlying digital information and computation. In the Urp, the digital shadow that accompanies the physical building models is such an example.

The success of a TUI often relies on a balance and strong perceptual coupling between the tangible and intangible representations. It is critical that both tangible and intangible representations be perceptually coupled to achieve a seamless interface that actively mediates interaction with the underlying digital information, and appropriately blurs the boundary between physical and digital. Coincidence of input and output spaces and real-time response are important requirements to accomplish this goal.

Note: There exist certain types of TUIs which have actuation of the tangible representation (physical objects) as the central mean of feedback. Examples are inTouch (Brave, Ishii, & Dahley, 1998), curlybot (Frei, Su, Mikhak, & Ishii, 2000), and topobo (Raffle, Parkes, & Ishii, 2004). This type of force-feedback-TUI does not depend on "intangible" representation since active feedback through the tangible representation serves as the main display channel.

Key Properties of TUI

While Fig. 24.2 illustrates the GUI's clear distinction between graphical representation and remote controls, the model of TUI illustrated in Fig. 24.3 highlights TUI's integration of physical representation and control. This model provides a tool for examining the following important properties and design requirements of tangible interfaces (Ullmer & Ishii, 2000).

Computational coupling of tangible representations to underlying digital information and computation. The central characteristic of tangible interfaces is the coupling of tangible representations to underlying digital information and computational models. One of the challenges of TUI design is how to map physical objects and their manipulation to digital computation and feedback in a meaningful and comprehensive manner.

As illustrated by the Urp example, a range of digital couplings and interpretations are possible, such as the coupling of data to the building models, operations to the wind tool, and property modifiers to the material wand.

Deciding the embodiment and mapping of the controller is dictated by the type of application envisioned. We give example cases in which a range of specificity of embodiment is used. In some applications, more abstract forms of physical objects (such as round pucks) are used as generic controllers that are reusable to control a variety of parameters by rotating and pushing a button (Patten, Ishii, Hines, & Pangaro, 2001). When a puck is used as a dial to control a simulation parameter, graphical feedback is given to complement the information, such as scale of the dial.

Embodiment of mechanisms for interactive control with tangible representations. The tangible representations of TUIs serve simultaneously as interactive physical controls. Tangibles may be physically inert, moving only as directly manipulated by a user's hands. Tangibles may also be physically actuated, whether through motor-driven force-feedback approaches (e.g., inTouch, Curlybot) or magnet-driven approaches such as Actuated Workbench (Pangaro, Maynes-Aminzade, & Ishii, 2002).

Tangibles may be unconstrained and manipulated in free space with six degrees of freedom. They may also be weakly constrained through manipulation on a planar surface, or tightly constrained, as in the movement of the abacus beads with one degree of freedom.

In order to make interaction simple and easy to learn, TUI designers need to utilize the physical constraints of the chosen physical embodiment. Because the physical embodiment, to some extent, limits the interaction choices, a designer must design the interaction so that the actions supported by the object are based on well-understood actions related to the physical object. For example, if a bottle shape is chosen, then opening the bottle by pulling out a cork is a well-understood mechanism (Ishii, Mazalek, & Lee, 2001). This understanding of the culturally common manipulation techniques helps disambiguate the users' interpretation of how to interact with the object.

Perceptual coupling of tangible representations to dynamic intangible representations. Tangible interfaces rely on a balance between tangible and intangible representations. Although embodied tangible elements play a central, defining role in the representation and control of a TUI, there is a supporting role for the TUI's intangible representation. A TUI's intangible representation—usually graphics and audio—often mediate much of the dynamic information provided by the underlying computation.

The real-time feedback of the intangible representation corresponding to the manipulation of the tangible representation is critical to insure perceptual coupling. The coincidence of inputs and output spaces (spatial continuity of tangible and intangible representations) is also an essential requirement to enhance perceptual coupling. For example, in Urp, the building models (tangible representation) are always accompanied by a "digital shadow" (intangible representation) without noticeable temporal or spatial gaps. That convinces users of an illusion that the shadows are cast by the building models (rather than by the video projector).

Genres of TUI Applications

By giving physical form to digital information to enhance an experience, TUIs have a wide variety of application domains. This section gives an overview of seven genres for promising TUI applications. For a more exhaustive survey of TUIs in a historical context, I would encourage the readers to refer to Ullmer and Ishii (2000), Holmquist, Redstr, and Ljungstrand (1999), and Fishkin (2004). Zuckerman, Arida, and Resnick (2005) also provided a useful taxonomy and frameworks to analyze the design space of TUIs.

Tangible Telepresence

One such genre is an interpersonal communication taking advantage of haptic interactions using mediated tangible representation and control. This genre relies on mapping haptic input to haptic representations over a distance. Also called "tangible telepresence," the underlying mechanism is the synchronization of distributed objects and the gestural simulation of "presence" artifacts, such as movement or vibration, which allow remote participants to convey their haptic manipulations of distributed physical objects. The effect is to give a remote user the sense of ghostly presence, as if an invisible person was manipulating a shared object. InTouch (Brave & Dahley, 1997), HandJive (Fogg, Cutler, Arnold, & Eisbach, 1998), and ComTouch (Chang, O'Modhrain, Jacob, Gunther, & Ishii, 2002) are examples of this.

Tangibles with Kinetic Memory

The use of kinesthetic gestures and movement to promote learning concepts is another promising domain. Educational toys to materialize, record, and play concepts have been also explored using actuation technology and taking advantage of i/o coincidence of TUI. Gestures in physical space illuminate the symmetric mathematical relationships in nature, and the kinetic motions can be used to teach children concepts relevant to programming and differential geometry as well as storytelling. Curlybot (Frei et al., 2000) and topobo (Raffle et al., 2004) are examples of toys which distill ideas relating gestures and form to dynamic movement, physics, and storytelling.

Constructive Assembly

Another domain is a constructive assembly approach that draws inspiration from LEGO™ and building blocks, building upon the interconnection of modular physical elements. This domain is mainly concerned with the physical fit between objects, and the kinetic relationships between these pieces that enable larger constructions and varieties of movement.

Constructive assembly was pioneered by Aish and Frazer in the late 1970s. Aish developed BBS (Aish, 1979; Aish & Noakes, 1984) for thermal performance analysis, and Frazer developed a series of intelligent modeling kits such as "Universal Constructor" (Frazer, 1994; Frazer, Frazer, & Frazer, 1980) for modeling and simulation. Recent examples include GDP (Anagnostou, Dewey, & Patera, 1989), AlgoBlock (Suzuki & Kato, 1993), Triangles (Gorbet, Orth, & Ishii, 1998), Blocks (Anderson et al., 2000), ActiveCube (Kitamura, Itoh, & Kishino, 2001), and System Blocks (Zuckerman & Resnick, 2004). Topobo (Raffle et al., 2004) is a unique instance that inherits the properties of both "constructive assemble" and "tangibles with kinetic memory."

Tokens and Constraints

"Tokens and constraints" is another TUI approach to operate abstract digital information using mechanical constraints (Ullmer, Ishii, & Jacob, 2005). Tokens are discrete, spatially reconfigurable physical objects that represent digital information or operations. Constraints are confining regions within which tokens can be placed. Constraints are mapped to digital operations or properties that are applied to tokens placed within their confines. Constraints are often embodied as physical structures that mechanically channel how tokens can be manipulated, often limiting their movement to a single physical dimension.

The Marble Answering Machine (Crampton Smith, 1995) is a classic example which influenced many following research. MediaBlocks (Ullmer, Ishii, & Glas, 1998), LogJam (Cohen, Withgott, & Piernot, 1999), DataTile (Rekimoto, Ulmer, & Oba, 2001),

and Tangible Query Interface (Ullmer, Ishii, & Jacob, 2003) are other recent examples of this genre of development.

Interactive Surfaces—Tabletop TUI

Interactive surfaces are another promising approach to support collaborative design and simulation which has been explored by many researchers in the past years to support a variety of spatial applications (e.g., Urp). On an augmented workbench, discrete tangible objects are manipulated and their movements are sensed by the workbench. The visual feedback is provided on the surface of the workbench, keeping input/output space coincidence. This genre of TUI is also called "tabletop TUI" or "tangible workbench."

Digital Desk (Wellner, 1993) is the pioneering work in this genre, and a variety of tabletop TUIs were developed using multiple tangible artifacts within common frames of horizontal work surface. Examples are metaDesk (Ullmer & Ishii, 1997), InterSim (Arias, Eden, & Fisher, 1997), Illuminating Light (Underkoffler & Ishii, 1998), Urp (Underkoffler & Ishii, 1999), Build-It (Rauterberg et al., 1998), Sensetable (Patten et al., 2001), AudioPad (Patten, Recht, & Ishii, 2002), and IP Network Design Workbench (Kobayashi, Hirano, Narita, & Ishii, 2003).

One limitation of above systems is the computer's inability to move objects on the interactive surfaces. To address this problem, the Actuated Workbench was designed to provide a hardware and software infrastructure for a computer to smoothly move objects on a table surface in two dimensions (Pangaro et al., 2002), providing an additional feedback loop for computer output, and helping to resolve inconsistencies that otherwise arise from the computer's inability to move objects on the table.

Continuous Plastic TUI

Fundamental limitation of previous TUIs was the lack of capability to change the forms of tangible representations during the interactions. Users had to use predefined finite sets of fixed-form objects, changing only the spatial relationship among them but not the form of individual objects themselves.

Instead of using predefined discrete objects with fixed forms, the new type of TUI systems utilizing continuous tangible material such as clay and sand were developed for rapid form-giving and sculpting for the landscape design. Examples are Illuminating Clay (Piper, Patti, & Ishii, 2002), and SandScape (Ishii, Ratti, Piper, Wang, Biderman, & Ben-Joseph, 2004). Later this interface was applied to the browsing of 3D volume metric data in PhoxelSpace project (Ratti, Wang, Piper, Ishii, & Biderman, 2004).

Augmented Everyday Objects

Augmentation of familiar everyday objects is an important design approach of TUI to lower the floor and to make it easy to understand the basic concepts. Examples are the Audio Notebook (Stifelman, 1996), musicBottles (Ishii et al., 1999), HandScape (Lee, Su, Ren, & Ishii, 2000), LumiTouch (Chang, Resner, Koerner, Wang, & Ishii, 2001), Designers' Outpost (Klemmer, Thomsen, Phelps-Goodman, Lee, & Landay, 2002), and I/O Brush (Ryokai, Marti, & Ishii, 2004). It is a challenge for industrial designers to improve upon a product by adding some digital augmentation to an existing digital object. This genre is open to much eager interpretation by artists and designers, to have our everyday physical artifacts evolve with technology.

Ambient Media

In the early stages of TUI research, we were exploring ways of improving the quality of interaction between people and digital information. We employed two approaches to extending interaction techniques to the physical world:

- Allowing users to "grasp and manipulate" foreground information by coupling bits with physical objects;
- Enabling users to be aware of background information at the periphery using ambient media in an augmented space.

At that time, HCI research had been focusing primarily on foreground activity on the screen and neglecting the rest of the user's computing environment (Buxton, 1995). However, in most situations, people are subconsciously receiving ambient information from their peripheral senses without attending to it explicitly. If anything unusual is noticed, it immediately comes to their attention, and they could decide to bring it to the foreground. For example, people subconsciously are aware of the weather outside their window. If they hear thunder, or a sudden rush of wind, the user can sense that a storm is on its way, out of his or her peripheral attention. If it was convenient, the user could then look outside, or continue working without distraction.

Ambient media describes the class of interfaces that is designed to smooth the transition of the users' focus of attention between background and foreground. Natalie Jeremijenko's Live Wire in 1995, at Xerox Parc, was a spinning wire that moved to indicate network traffic. Designing simple and adequate representations for ambient media using tangible objects is a key part of the challenge of Tangible Bits (Ishii & Ullmer, 1997).

The ambientROOM is a project that explores the ideas of ambient media constructing a special room equipped with embedded sensors and ambient displays (Ishii et al., 1998). This work was a preliminary investigation into background/peripheral interfaces, and led to the design of standalone ambient fixtures such as Pinwheels and Walter Lamps that make users aware of "digital wind" and "bits of rain" at their peripheral senses (Dahley, Wisneski, & Ishii, 1998).

Strictly speaking, ambient media is not a kind of TUI since in many cases there are no direct interactions. Rather, ambient media serve as background information displays that complement tangible/graspable media that users manipulate in the foreground. TUI's approach to ambient media is concerned with the design of simple mappings that give easy-to-understand form to cyberspace information and represent change in a subtle manner. We started experimenting with a variety of ambient media such as sound, light, airflow, and water movement for background interfaces for awareness of cyberspace at the periphery of human perception.

This concept of "ambient media" is now widely studied in the HCI community as a way to turn the architectural/physical spaces into an ambient and calm information environment. Another design space is low-attention interfaces for interpersonal communication through ambient media (Chang et al., 2001). Ambient devices further commercialized the domain of low-attention ambient media interfaces by developing the Ambient Orb and Weather Beacon, exploring the new genre of "glanceable interfaces" (http://www.ambientdevices.com/).

TUI Instances

In this section, 10 TUI examples are presented to illustrate the potential application domains described in a previous section, and to highlight unique features of TUIs. However, given the limited space and rapid growth of TUI research in HCI community in recent years, the collection of examples introduced here can only cover a relatively small portion of the representative works of TUIs.

InTouch: Tangible TelePresence Through Distributed Synchronized Physical Objects

InTouch is a project to explore new forms of interpersonal communication over distance through touch by preserving the physical analog movement of synchronized distributed rollers (Brave & Dahley, 1997; Brave et al., 1998). Force-feedback is employed to create the illusion that people, separated by distance, are interacting with a shared physical object. The "shared" object provides a haptic link between geographically distributed users, opening up a channel for physical expression over distance.

Two identical mechanisms were built with three freely rotating rollers (Photo 24.3). Each roller is synchronized to the corresponding roller on the distant mechanism using force-feedback, so that when one roller is moved the other corresponding roller also moves. If the movement of one roller is held, then the roller transmits that resistance to the other roller. They are in a sense connected by a stiff computational spring. Two users separated by distance can then play, moving or tapping the rollers or more passively feeling the other person's manipulation of the object. The presence of the other person is represented tangibly through physical interaction with the inTouch device.

Force-feedback is conventionally used to allow a user to "touch" virtual objects in the computer screen through a single point. InTouch applies this technology to realize a link for interpersonal haptic communication, instead of just touching virtual objects. InTouch allows people to feel as if they are connected through touching the rollers, to another person. Instead of touching inanimate objects, each person is touching a dynamic, moving object that is shared.

Important features of inTouch from HCI points of view can be summarized as follows:

1. No boundary between "input" and "output" (i/o coincidence: the wooden rollers are force displays as well as input devices);
2. Principal human input/output organs are hands, not eyes or ears (with the sense of touch being the primary mode);
3. Information can be sent and received simultaneously through one's hand.

Past communication media such as video-telephony set themselves the ultimate goal of reproducing the voice or the image of the human face and body as realistically as possible in order to create the illusion of "being there" for each interlocutor. InTouch takes the opposite approach by making users aware of the other person without ever rendering him or her in bodily terms, and creating what we call a "tangible presence" or "ghostly presence." By seeing and feeling an object being moved in a human fashion on its own, we imagine a ghostly body. The concept of the ghostly presence provides us with a different approach to the conventional notion of telepresence.

Curlybot: A Toy to Record and Play

Curlybot is a toy that can record and play back physical motion (Photo 24.4). As one plays with it, it remembers how it has been moved and can replay that movement with all the intricacies of the original gesture; every pause, acceleration, and even the shaking in the user's hand, is recorded. Curlybot then repeats that gesture indefinitely, creating beautiful and expressive patterns. Children can use curlybot to gain strong intuition for advanced mathematical and computational concepts, like differential geometry, through play outside of traditional computers (Frei et al., 2000).

The forced-feedback technology used for real-time simultaneous communication in inTouch was employed in curlybot for the recording and playback of gestures. Two motors equipped with an optical encoder enable free rotation in addition to forward and backward movement.

When the user presses the button a red LED is illuminated to indicate the recording mode. The user then moves the curlybot around; meanwhile an encoder is recording this gesture information. Pushing the button a second time terminates recording and a green LED alights to indicate the playback mode. The microprocessor compares the current position with the stored positions and instructs the motors to retrace the steps recorded in the curlybot's memory.



PHOTO 24.4. Curlybot.

This project contributes to both interface design and education. As a tangible interface it blurs the boundary between input and output, as inTouch does. Curlybot itself is both an input device to record gestures and a physical display device to reenact them. By allowing the user to teach it gestures with his or her hand and body, and then reenacting those gestures in a physical space around the body, curlybot enables a strong connection between body and mind not obtainable from anything expressed on a computer screen.

From an educational standpoint, curlybot allows very young children to explore "advanced" mathematical and computational concepts. Curlybot supports new ways of thinking about geometric shapes and patterns. Children can also use curlybot to explore some of the basic ideas behind computational procedures, like how complexity can be built from simple parts. This is similar to what is possible with the Logo programming language, but does not require children to read or write and thus makes advanced ideas accessible to younger children. Curlybot also draws strongly on children's intuition about their own physical actions in the world to learn—what Seymour Papert called "body syntonic learning" (Papert, 1980). In addition, the direct input and beautifully expressive patterns that result through curlybot's repetition of the gestures keep children playing and engaged.

Topobo: 3D *Constructive Assembly With Kinetic Memory*

Topobo, a combination of "topology" and "robotics," is a 3D constructive assembly system with kinetic memory, and the ability to record and play back physical motion (Raffle et al., 2004). By snapping together a combination of passive (static) and active (motorized) components, people can quickly assemble dynamic biomorphic forms like animals and skeletons with Topobo. Topobo allows users to animate those forms by recording the movement of pushing, pulling, and twisting them, and later observe the system play back those motions repeatedly. This "record and play" function was inherited from the prior curlybot project, and the constructive assembly function was inherited from the commercial toy, Zoob™.

For example, a dog can be constructed and then taught to gesture and walk by twisting its body and legs. The dog will then repeat those movements and walk repeatedly. The same way people can learn about static structures playing with regular building blocks, they can learn about dynamic structures by playing with Topobo. Topobo works like an extension of the body, giving one gestural fluency. Topobo embeds computation within a dynamic building system so that gestural manipulation of the material becomes a programming language (Photo 24.5).

Topobo was inspired by current trends in computational media design and by artists and empiricists using visual explorations and models of natural phenomena to more deeply appreciate patterns found in the natural world. In this spirit, Topobo is designed to allow people to use experimentation, play, and self-expression to discover and explore common natural relationships between natural forms and dynamic motion. Building toys and educational manipulatives have been used for years by children to learn about the world though model making.

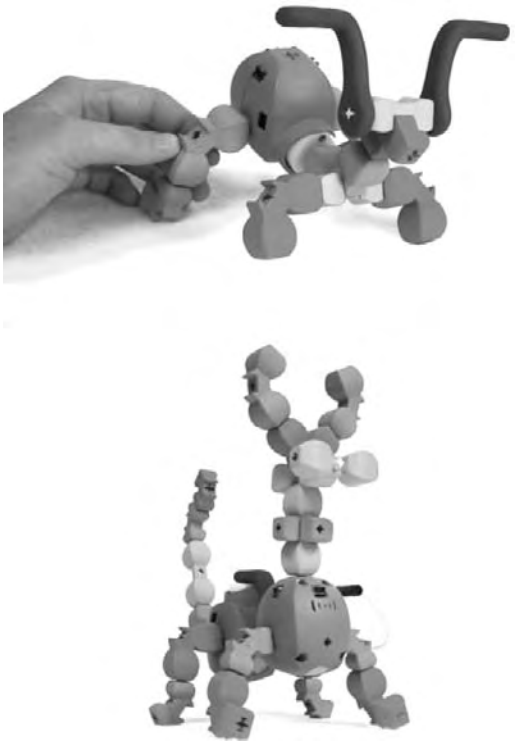


PHOTO 24.5. Topobo.

Unique among modeling systems is Topobo's coincident physical input and output behaviors (which is also common to inTouch and curlybot). The system is comprised of 10 different primitives that can be snapped together in a variety of ways. Nine of these primitives are called "passive" because they form static connections. These static connections constrain the form and the range of motion available to the structure. One "active" primitive is built with an embedded encoder and motor which is programmed by demonstration. These motorized components are the only ones that move, so the system is able to faithfully record and replay every dynamic manipulation to the structure.

*media*Blocks: *Token and Constraint Approach*

The mediaBlocks system is a tangible interface for manipulating lists of online digital media such as video clips and images (Ullmer et al., 1998). Whereas Urp provides a spatial interface for leveraging object arrangements consistent with real-world building configurations, the mediaBlocks system provides a relational interface for manipulating more abstract digital information.

The mediaBlocks are small, digitally tagged blocks, dynamically bound to lists of online media elements. The mediaBlocks support two major modes of use. First, they function as capture, transport, and playback mechanisms for moving online media between different media devices. In this mode, conference room cameras, digital whiteboards, wall displays, printers, and other devices are outfitted with mediaBlock slots. Inserting one of the mediaBlocks into the slot of a recording device (e.g., a

camera) activates the recording of media into online space, and the dynamic binding of the media to the physical block.

Similarly, inserting one of the bound mediaBlocks into a playback device (e.g., video display) activates playback of the associated online media. Inserting mediaBlocks into slots mounted on computer monitors provides an intermediate case, allowing mediaBlock contents to be exchanged bi-directionally with traditional computer applications using the GUI drag-and-drop operation.

The second functionality of mediaBlocks uses the blocks as physical controls on a media-sequencing device (Photo 24.6). A mediaBlock "sequence rack" (partially modeled after the tile racks of the Scrabble game) allows the media contents of multiple adjacent mediaBlocks to be dynamically bound to a new mediaBlock carrier. Similarly, a second "position rack" maps the physical position of a block to an indexing operation upon its contents. When mediaBlocks are positioned on the left edge of the position rack, the first media element of the block is selected. Intermediate physical positions on the rack provide access to later elements in the associated media list of the block.

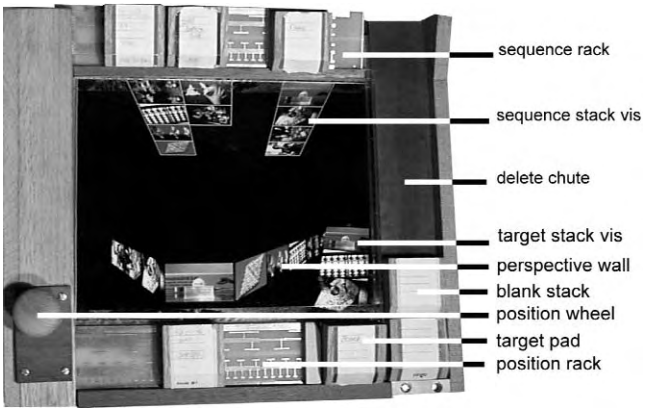


PHOTO 24.6. mediaBlocks: media sequencing device.



PHOTO 24.7. AudioPad running on Sensetable platform.

Digital Desk: *Pioneer of Tabletop* TUI

Digital Desk (Wellner, 1993) is a pioneering work to demonstrate a way to integrate physical and digital document processing on a table. Wellner brought some of the functionality we typically associate with GUIs onto the physical desktop. This table used a camera and a microphone to detect finger presses on a graphical interface displayed on a desk with a video projector. Wellner used this desk for tasks such as graphic design and spreadsheet computations on physical paper. This system also employed some physical props, such as a scanner that would scan items and place them directly on the tabletop interaction surface.

Wellner's research pointed the way toward enabling the computer to perform some of the operations we traditionally associate with GUIs in a tabletop environment. The Digital Desk also illustrates some of the compelling reasons for considering computer interfaces based on horizontal interactive surfaces. Because many work surfaces in our environment are already planar, horizontal or nearly horizontal surfaces, integrating computer interfaces into these surfaces may provide an opportunity for new types of relationships between computation and physical objects, and may help create computer systems that are more relevant to problem domains with established work practices based on tabletops.

The Digital Desk inspired many tabletop tangible interfaces including the Luminous Room project (Underkoffler, Ullmer, & Ishii, 1999) from which Urp (Underkoffler & Ishii, 1999) was created. Sensetable (Patten et al., 2001) is another example.

Sensetable and AudioPad: *Tabletop* TUI *for Real-Time Music Performance*

Sensetable (Patten et al., 2001) is a system that wirelessly tracks the positions of multiple objects on a flat display surface. The Sensetable serves as a common platform for a variety of tabletop TUI applications such as Audio Pad and IP Network Design Workbench.

Audiopad (Patten et al., 2002) is a composition and performance instrument for electronic music which tracks the positions of objects on a tabletop surface and converts their motion into music. One can pull sounds from a giant set of samples, juxtapose archived recordings against warm synthetic melodies, cut between drum loops to create new beats, and apply digital processing all at the same time on the same table. Audiopad not only allows for spontaneous reinterpretation of musical compositions, but also creates a visual and tactile dialogue between itself, the performer, and the audience.

Audiopad is based on the Sensetable platform that has a matrix of antenna elements which track the positions of electronically tagged objects on a tabletop surface. Software translates the position information into music and graphical feedback on the tabletop. Each object represents either a musical track or a microphone (Photo 24.8).

Experience of Audiopad with tangible user interface through a series of live performances suggests that interacting with electromagnetically tracked objects on a tabletop surface with graphical feedback can be a powerful and satisfying tool for musical expression. The integration of input and output spaces gives the



PHOTO 24.8. IP network design workbench running on Sensetable platform.

performer a great deal of flexibility in terms of the music that can be produced. At the same time, this seamless integration allows the performer to focus on making music, rather than using the interface. Spatial multiplexed inputs of TUI also supported two performers play music simultaneously and collaboratively (Photo 24.8).

IP *Network Design Workbench*: *Event Driven Simulation on Sensetable*

The IP Network Design Workbench (IPNWDWB) is the collaborative project between NTT Comware and the Tangible Media Group. The IP Network Design Workbench supports collaborative network design and simulation by a group of experts and customers (Kobayashi et al., 2003). This system is also based on the Sensetable platform which can wirelessly detect the location and orientation of physical pucks. Simulation engine is based on the event-driven simulation model. Using the Sensetable system, users can directly manipulate network topologies for modeling, control simulation parameters of nodes and links using physical pucks on the sensing table, and simultaneously see the simulation results projected onto the table in real time (Photo 24.9).

The goal of IPNWDWB is to make simulation tools more accessible for non-experts, so that they can join the network design process and interact with experts more easily than using traditional GUI computer. This system was commercialized and has been used for collaborative network design with customers to ensure their understanding of the performance and cost of network enhancements dealing with the increases of network traffic caused by Voice over IP and/or streaming video, for example. Because of the large tiling horizontal work surface and TUI interaction that invites all the participants to touch and manipulate pucks simultaneously, the process of decision making becomes much more democratic and more convincing than ordinary PowerPoint presentations through conventional GUI.

If we compare IPNWDWB with Urp, we notice a big difference in the nature of applications. In Urp, we used physical scale models of buildings, which humans have used for thousands of years to design cities, as tangible representations of urban models. Therefore, it is very natural to apply TUIs to such domains (e.g., urban planning, landscape design) in which physical models have been used long before the birth of digital computers.

In contrast, IP Network Design is based on event-driven simulation models which are quite abstract and new. This modeling technique requires digital computers. IPNWDB is important

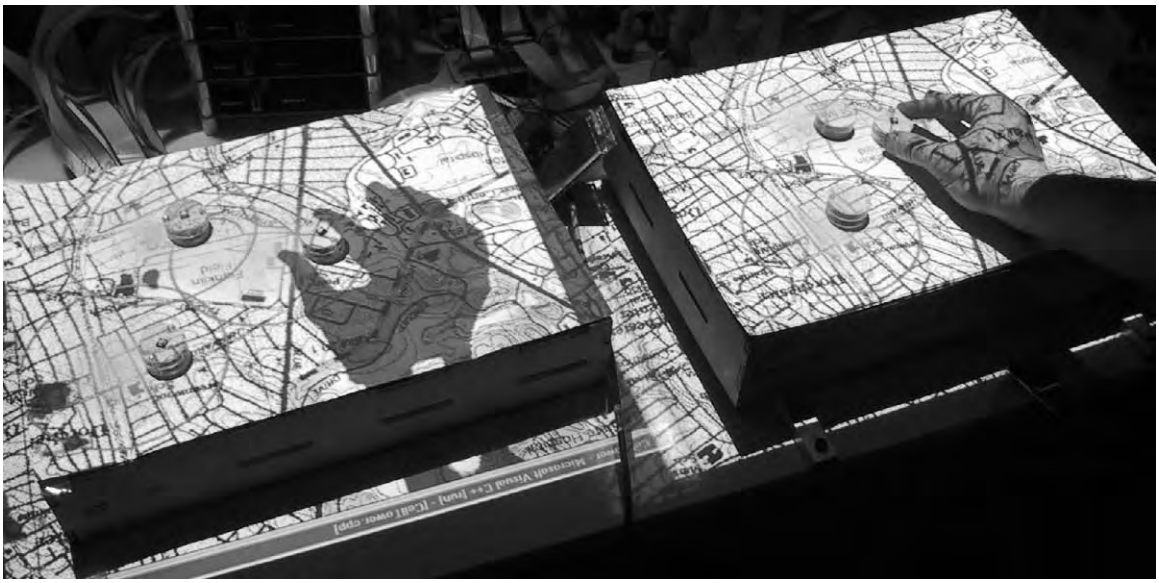


PHOTO 24.9. Actuated workbench used for distributed collaboration.

since it demonstrated that TUI can empower the design process even in abstract and computational application domain which does not have straight-forward mappings from abstract concepts to physical objects. There are a wide range of modeling and simulation techniques such as System Dynamics and Event-Driven Simulation that uses 2D graph representation. We learned that many of these abstract computational applications can be supported by Sensetable-like TUI platforms in the collaborative design sessions. For example, simultaneously changing parameters, transferring control between different people or different hands, and distributing the adjustment of simulations dynamically are interactions enabled by TUI.

Actuated Workbench: Closing a Loop of Computational Actuation and Sensing

The aforementioned tabletop TUI systems share a common weakness. While input occurs through the physical manipulation of tangible objects, output is displayed only through sound or graphical projection on and around the objects. As a result, the objects can feel like loosely coupled handles to digital information rather than physical manifestations of the information itself.

In addition, the user must sometimes compensate for inconsistencies when links between the digital data and the physical objects are broken. Such broken links can arise when a change occurs in the computer model that is not reflected in a physical change of its associated object. With the computer system unable to move the objects on the table surface, it cannot undo physical input, correct physical inconsistencies in the layouts of the objects, or guide the user in the physical manipulation of the objects. As long as this is so, the physical interaction between human and computer remains one-sided.

To address this problem, the Actuated Workbench was designed to provide a hardware and software infrastructure for a computer to smoothly move objects on a table surface in two dimensions (Pangaro et al., 2002).

The Actuated Workbench is a new technology that uses magnetic forces to move objects on a table in two dimensions. It is intended for use with existing tabletop tangible interfaces, providing an additional feedback loop for computer output, and helping to resolve inconsistencies that otherwise arise from the computer's inability to move objects on the table.

Actuation enables a variety of new functions and applications. For example, a search and retrieve function could respond to a user query by finding matching data items and either moving them to another place on the tabletop or wiggling them to get the user's attention. A more powerful function would be one in which the computer could physically sort and arrange pucks on the table according to user-specified parameters. This could help the user organize a large number of data items before manually interacting with them. As a user makes changes to data through physical input, he or she may wish to undo some changes. A physical undo in this system could move the pucks back to their positions before the last change. It could also show the user the exact sequence of movements she had performed. In this sense, both "undo" and "rewind" commands are possible.

One advantage that tabletop tangible user interfaces offer is the ease with which multiple users can make simultaneous changes to the system. Users can observe each other's changes, and any user can reach out and physically change the shared layout without having to grab a mouse or other pointing device. This is not the case, however, when users are collaborating remotely. In this scenario, a mechanism for physical actuation of the pucks becomes valuable for synchronizing multiple, physically separated workbench stations (Photo 24.9). Without such a mechanism, real-time physical synchronization of the two tables would not be possible, and inconsistencies could arise between the graphical projection and the physical state of the pucks on the table.

In addition to facilitating the simple synchronization of these models, the Actuated Workbench can recreate remote users' actual gestures with objects on the table, adding greatly to the "Ghostly Presence" (Brave et al., 1998) sought in remote-collaboration interfaces.

Actuated Workbench is helpful in teaching students about physics by demonstrating the attraction and repulsion of charged particles represented by pucks on the table. As a student moves the pucks around on the table, the system could make them rush together or fly apart to illustrate forces between the objects.

SandScape: Continuous TUI for Landscape Design

SandScape (Ishii et al., 2004) is a tangible interface for designing and understanding landscapes through a variety of computational simulations using sand. Users view these simulations as they are projected on the surface of a sand model that represents the terrain. The users can choose from a variety of different simulations that highlight the height, slope, contours, shadows, drainage, or aspect of the landscape model (Photo 24.10).

The users can alter the form of the landscape model by manipulating sand while seeing the resultant effects of computational analysis generated and projected on the surface of sand in real time. The project demonstrates how TUI takes advantage of our natural ability to understand and manipulate physical forms while still harnessing the power of computational simulation to help in our understanding of a model representation.

The SandScape configuration is based on a box containing 1 m-diameter glass beads lit from beneath with an array of 600 high-power infrared LEDs. Four IR mirrors are placed around the LED array to compensate for the uneven radiance distribution on the boundary. A monochrome infrared camera is mounted 2 m above the surface of the beads and captures the intensity of light passing through the volume. The intensity of transmitted light is a function of the depth of the beads and a look-up table can be used to convert surface radiance values into surface elevation values. The system has been calibrated to work with a specific bead size and the optical properties of the material used (absorption and scattering coefficients) are critical to its successful functioning. Owing to the exponential decay of the IR light passing through the glass beads (or any other material) the intensity at the top surface can vary greatly and sometimes exceed the dynamic range of the video camera. This problem can be solved by taking several images with different exposure times

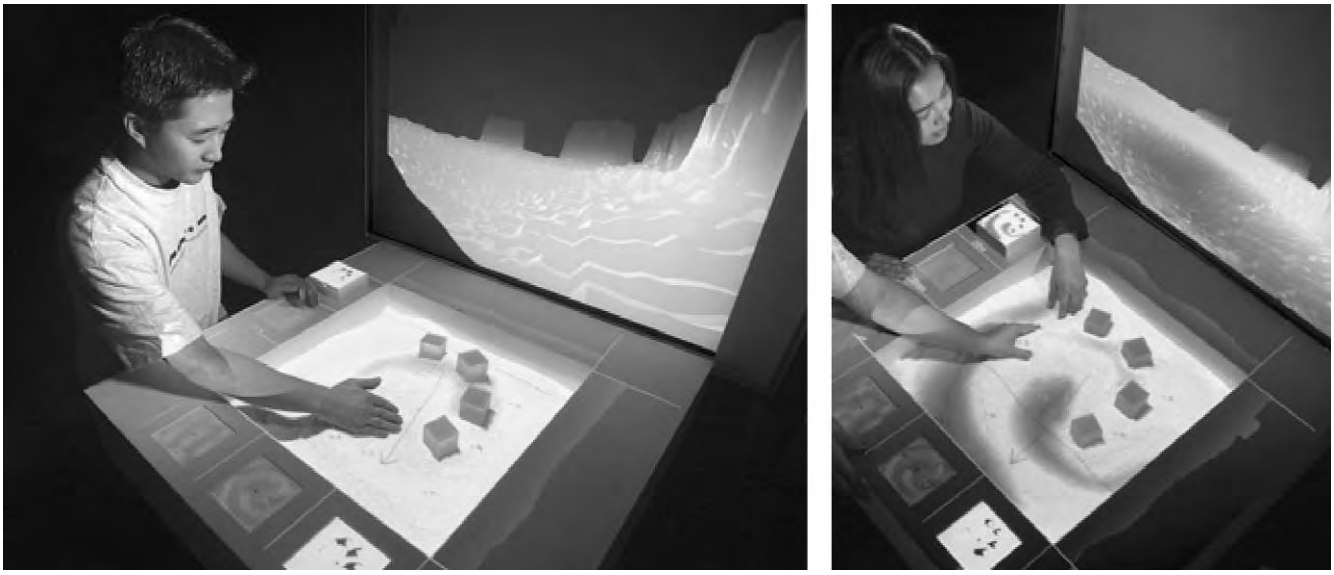


PHOTO 24.10. SandScape.

and combining them to recover the effective radiance of the scene. SandScape is less accurate than its predecessor Illuminating Clay, which used laser rangefinders to capture the geometry of a clay model (Piper et al., 2002).

SandScape and Illuminating Clay show the potential advantages of combining physical and digital representations for landscape modeling and analysis. The physical clay and sand models convey spatial relationships that can be intuitively and directly manipulated by hand. Users can also insert any found physical objects directly under the camera. This approach allows users to quickly create and understand highly complex topographies that would be difficult and time-consuming to produce with conventional CAD tools. We believe that this "Continuous TUI" approach makes better use of our natural abilities to discover solutions through the manipulation of physical objects and materials.

At the same time, the projected graphics give the user real-time feedback. While tracked physical models interfaced with a computer are not a novelty, we believe that SandScape and Illuminating Clay offer a new contribution, by using the continuous surface geometry of the model itself to act as the input/output mechanism. In so doing we hope to give the projected information the same tangible immediacy as the clay/sand material itself and allow quantitative data to support the intuitive understanding of the landscape.

Landscape architecture, as well as urban and architectural design, requires the collaboration of a number of specialists. These include earth engineers, water engineers, agrarian managers, land economists, and transport engineers—to name just a few. In the current process of design, the collaboration happens at different stages, and sometimes without much direct and synchronous interaction. SandScape and Illuminating Clay provide a common platform for collaboration, centered on the table workspace. Numerous representations and analyses can be combined in a single design environment, potentially offering a greater cohesion between different specialists and streamlining the process of design.

musicBottles: Transparent Interface Based on Augmented Glass Bottles

musicBottles introduces a tangible interface that deploys bottles as containers and controls for digital information (Photo 24. 11). The system consists of a specially designed table and three corked bottles that "contain" the sounds of the violin, the cello, and the piano in Edouard Lalo's Piano Trio in C Minor, Op. 7. Custom-designed electromagnetic tags embedded in the bottles enable each one to be wirelessly identified.

When a bottle is placed onto the stage area of the table, the system identifies each bottle, and lights up the stage to show that the bottles have been recognized. The opening and closing of a bottle is also detected, and as the cork is removed, the corresponding instrument becomes audible. A pattern of colored light is rear-projected onto the table's translucent surface to reflect changes in pitch and volume for each instrument. The interface allows users to structure the experience of the musical composition by physically manipulating the different soundtracks.

Humans have used glass bottles for thousands of years. Through the seamless extension of physical affordances and metaphors of the bottles into the digital world, the bottles project explores the transparency of the interface (Ishii, 2004).

A wide variety of contents, including music, weather reports, poems, and stories have been designed to test the concept (Ishii et al., 1999). The bottles lined up on a specially designed table, the feel of the glass as we open them, and the music and light from the LED lamps that come out of them together create a unique aesthetic experience. This is a pleasure not to be had from the mere click of a mouse.

Potential applications are not limited to music alone. One might imagine perfume bottles filled with poetry or wine bottles that decant stories (Mazalek, Wood, & Ishii, 2001). More practical applications might include a medicine chest full of bottles that tell the user how and when to take them and let the hospital

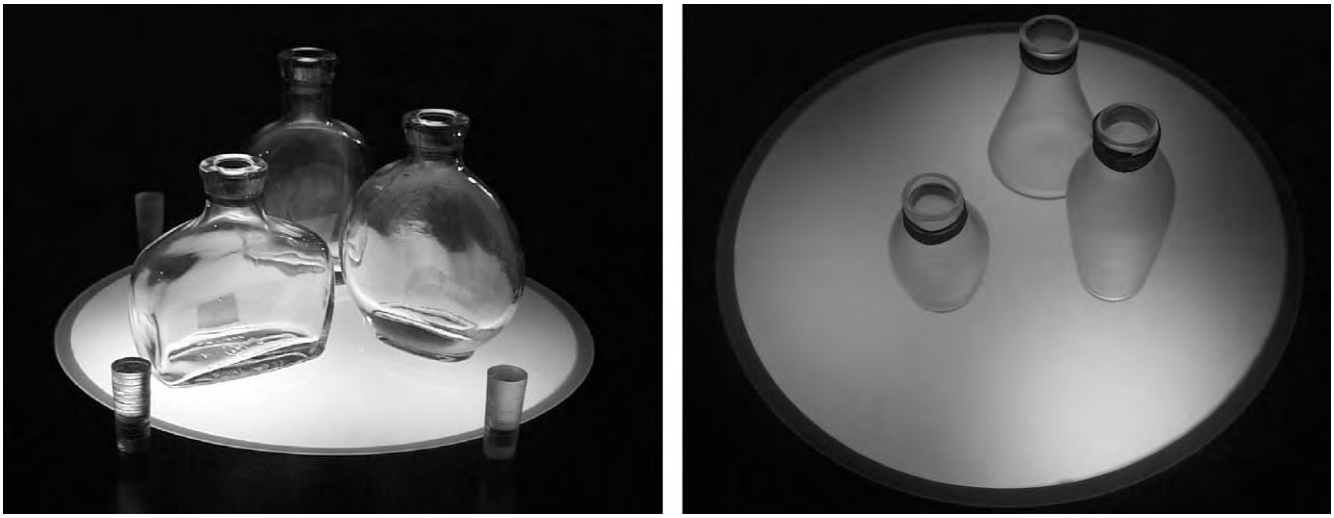


PHOTO 24.11. musicBottles.

know when they do. As an intimate part of our daily lives, glass bottle interfaces offer a simple and transparent interface.

Pinwheels: Ambient Interface to Information

Pinwheels are an example of ambient media that demonstrate a new approach to interfacing people with online digital information through subtle changes in sound and movement, which can be processed in the background of awareness. Pinwheels spin in a "bit wind" and represent an invisible flow of digital information such as network traffic as physical movement within an architectural spaces (Photo 24.12).

Nature is filled with subtle, beautiful, and expressive ambient media that engage each of our senses. The sounds of rain and the feeling of warm wind on our cheeks help us understand and enjoy the weather even as we engage in other activities. Similarly, we are aware of the activity of neighbors through passing sounds and shadows at the periphery of our attention. Cues like an open door or lights in an office help us subconsciously understand the activities of other people, and communicate our own activity and availability.

Current personal computing interfaces, however, largely ignore these rich ambient spaces, and squeeze vast amounts of digital information into small rectangular screens. Information is presented as "painted bits" (pixels) on flat screens that must be in the center (foreground) of a user's focus to be processed. In order to broaden the concept of "display" to make use of the entire physical environment as an interface, using ambient media, information can be manifested as subtle changes in form, movement, sound, color, smell, temperature, or light. We call them "ambient displays."

The Pinwheels evolved from the idea of using airflow as ambient media. However, we found that the flow of air itself was difficult to control and to convey information. As an alternative, we envisioned that a visual/physical representation of airflow based on the "spinning pinwheels" could be legible and poetic.



PHOTO 24.12. Pinwheels.

The Pinwheels spin in the "bit wind" at different speeds based upon their input information source.

Ambient displays are envisioned as being all around and suited to the display of

1. People's presence (awareness of remote people's status/activities);
2. Atmospheric and astronomical phenomena;
3. General states of large and complex systems (e.g., an atomic power plant).

For instance, an atmospheric scientist might map patterns of solar wind into patterns of Pinwheel spins in a room.

There are many design challenges surrounding ambient displays. One of them is the mapping of information to the physical motion and other ambient media. A designer of ambient displays must transform the digital data into a meaningful pattern

of physical motion that successfully communicates the information. The threshold between foreground and background is another key issue. Ambient displays are expected to go largely unnoticed until some change in the display or user's state of attention makes it come into the foreground of attention. How to keep the level of display at the threshold of a user's attention is an open design issue.

Contributions of TUIs

TUI is generally built from systems of physical artifacts with digital coupling with computation. Taken together as ensembles, TUI has several important advantages over traditional GUI as well as limitations. This section summarizes those contributions of TUIs and required design considerations.

Double Interactions Loop: Immediate Tactile Feedback

One important advantage of TUI is that users receive passive haptic feedback from the physical objects as they grasp and manipulate them. Without waiting for the digital feedback (mainly visual), users can complete their input actions (e.g., moving a building model to see the interrelation of shadows).

Typically there are two feedback loops in TUI, as shown in Fig. 24.4:

1. The passive haptic feedback loop provides the user with an immediate confirmation that he or she has grasped and moved the object. This loop exists within a physical domain, and it does not require any sensing and processing by a computer. Thus, there is no computational delay. The user can begin manipulating the object as desired without having to wait for the second feedback loop, the visual confirmation from the interface. In contrast, when user uses a mouse with a GUI computer, he or she has to wait for the visual feedback (second loop) to complete an action.
2. The second loop is a digital feedback loop that requires sensing of physical objects moved by users, computation based on the sensed data, and displaying the results as visual (and auditory) feedback. Therefore, this second loop takes longer than the first loop.

Many of the frustrations of using current computers come from the noticeable delay of digital feedback as well as a lack of tactile confirmation of actions taken by computers. We believe the double loops of TUI give users a way to ease those frustrations.

Note: Actuation technology introduced in Actuated Workbench will contribute to add another loop, that of physical actuation. Figure 24.5 illustrates the third loop introduced into the TUI model by computer-controlled actuation and sensing. The third loop allows the computer to give feedback on the status of the digital information as the model changes or responds to internal computation.

Persistency of Tangibles

As physical artifacts, TUI tangibles are persistent. Tangibles also carry physical state, with their physical configurations tightly coupled to the digital state of the systems they represent. The physical state of tangibles embodies key aspects of the digital state of an underlying computation.

For example, the physical forms of the Urp building models, as well as their position and orientation on the workbench of the system, serve central roles in representing and controlling the state of the underlying digital simulation system. Even if the mediating computers, cameras, and projectors of Urp are turned off, many aspects of the state of the system are still concretely expressed by the configuration of its physical elements.

In contrast, the physical form of the mouse holds little representational significance because GUIs represent information almost entirely in visual form.

Coincidence of Input and Output Spaces

Another important feature (and design principle) of TUI is coincidence of input and output spaces to provide seamless information representation that spans both tangible (physical) and intangible (digital) domains.

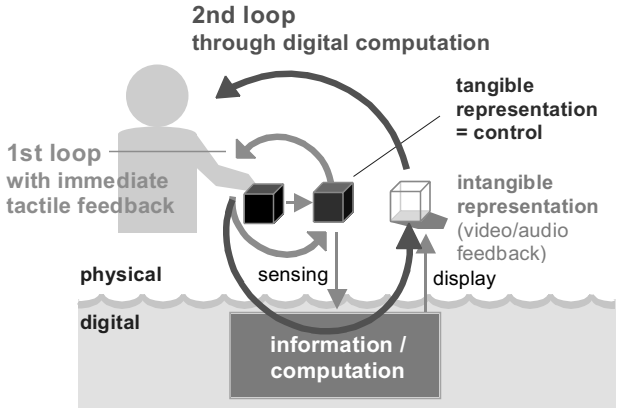


FIGURE 24.4. TUI's double feedback loops.

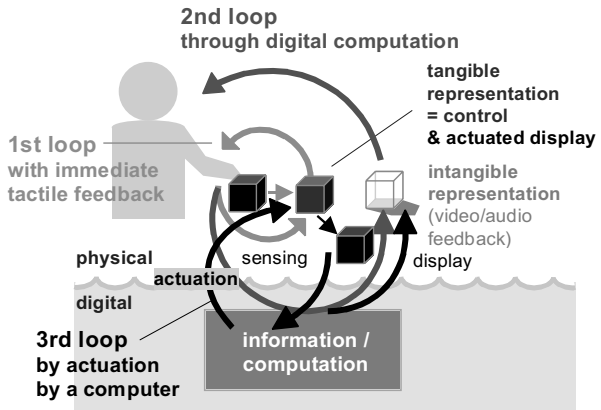


FIGURE 24.5. TUI with actuation (Actuated workbench).

GUI utilizes the mouse and keyboard as generic "remote" controllers (input), and the screen serves as main output medium. Thus, there is spatial discontinuity between those two spaces. There is also multimodal inconsistency, as touch is the main input while vision is the only output.

TUI tries to coincide inputs space and output space as much as possible to realize seamless coupling of physical and digital worlds (Ishii & Ullmer, 1997). An example of this seamless coupling of is Underkoffler's Urp (Underkoffler & Ishii, 1999). A series of architectural models serve as the input devices, and output in the form of a wind-and-shadow simulation is projected down onto the same tabletop surface, on top of and around the building models. Illuminating Clay (Piper et al., 2002) and SandScape (Ishii et al., 2004) demonstrate another example of i/o coincidence using continuous flexible material: sand. Curlybot and topobo demonstrate the same concept using the contact surface of the tangibles as input and output to digitize the person's physical motion.

Special Purpose vs. General Purpose

GUIs are fundamentally general-purpose interfaces that are supposed to emulate a variety of applications visually using dynamic pixels on a screen and generic remote controllers such as the mouse and keyboard. On the other hand, TUIs are relatively specific interfaces tailored to certain type of applications in order to increase the directness and intuitiveness of interactions.

The selection of the correct and specific application domain is critical to apply TUI successfully to take advantage of existing skills and work practices (e.g., use of physical models in urban planning).

One notable aspect of Urp is its use of objects with very application-specific physical forms (scaled building models) as a fundamental part of the interface. Physical building models represent the buildings themselves in the interactive simulation. Thus they give the user important visual and tactile information about the computational object they represent. Indicators such as a clock and weather vane work in reverse in the Urp system. Instead of the clock hands moving to indicate the passage of time, the user can move the clock hands to change the time of day for the shadow study (Photo 24.1). Likewise, he or she can change the orientation of the weather vane to control the direction of the wind (Photo 24.2).

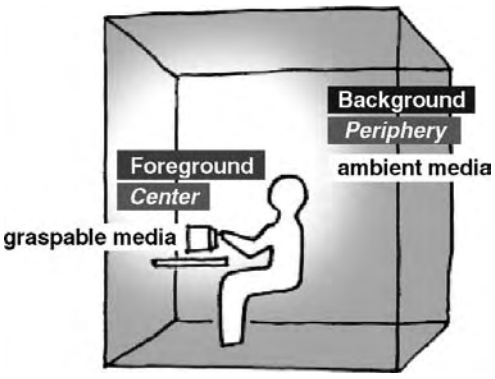


FIGURE 24.6. Center and periphery of user's attention within physical space.

In the design of TUI, it is important to give an appropriate form to each tangible tool and object so that the form will give an indication of the function available to the users. For example, the clock hands allow people to automatically make the assumption that they are controlling time.

Of course, this special-purpose-ness of TUIs can be a big disadvantage if users would like to apply it to a wide variety of applications since customized physical objects tailored to certain application can not be reused for most of other applications. By making the form of objects more abstract (e.g., a round puck), you lose the legibility of tangible representation and the object will become a generic handle rather than the representation of underlying digital information. It is important to attain a balance between specific/concrete vs. generic/abstract to give a form to digital information and computational function.

Space-Multiplexed Input

Another distinct feature of TUI is space-multiplexed input (Fitzmaurice, Ishii, & Buxton, 1995a). Each tangible representation serves as a dedicated controller occupying its own space, and encourages two-handed and multi-user simultaneous interaction with underlying computational model. Thus, TUI is suitable for collocated collaboration allowing concurrent manipulation of information by multiple users.

GUI, in contrast, provides time-multiplexed input that allows users to use one generic device to control different computational functions at different points in time. For instance, the mouse is used for menu selection, scrolling windows, pointing, and clicking buttons in a time-sequential manner.

TUI can support not only collocated collaboration, but also remote collaboration using actuation mechanism to synchronize the physical states of tangibles over distance. Actuated Workbench is an example of such a technology that extends TUI for remote collaboration (Pangaro et al., 2002).

In the Urp scenario, applying the Actuated Workbench technology, it is possible to have two distributed Urp tables in different locations, connected and synchronized over the Internet. One Urp can be in Tokyo, while the other Urp can be in Boston, and the shadows are synchronized as the urban planning team moves the buildings around the Urp space. The movement of buildings can be also synchronized by the actuation mechanism. When the building planner moves a building location, both the local and the remote shadow will update simultaneously; position and orientation of moved buildings is also synchronized. This synchronization of a distributed workbench allows both teams to discuss changes to the situation in real time, and provides a common reference for otherwise ethereal qualities such as wind, time, and shadow.

CONCLUSION

The author met a highly successful computational device called the "abacus" when he was two years old (Photo 24.13). He could enjoy the touch and feel of the "digits" physically represented as

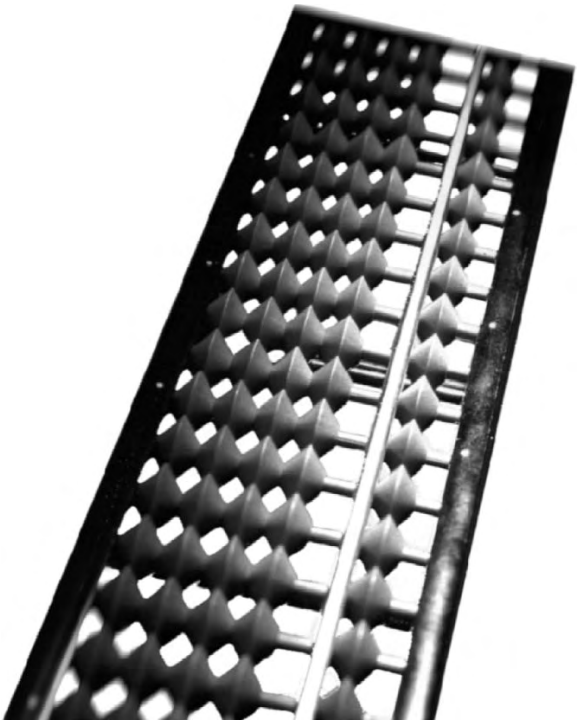


PHOTO 24.13. Abacus.

arrays of beads. This simple abacus was not merely a digital computational device. Because of its physical affordance, the abacus also became a musical instrument, imaginary toy train, and a backscratcher. He was captivated by the sound and tactile interaction with this simple artifact.

His childhood abacus became a medium of awareness too. When his mother kept household accounts, he was aware of her activities by the sound of her abacus, knowing he could not ask her to play with him while her abacus made its music.

This abacus suggests to us a new direction of Human-Computer Interaction (HCI) that we call Tangible User Interfaces (TUI). First, it is important to note that the abacus makes no distinction between "input" and "output." Instead, the beads, rods, and frame serve as physical representations of numerical information and computational mechanism. They also serve as directly manipulatable physical controls to compute on numbers.

Second, the simple and transparent mechanical structure of the abacus (without any digital black boxes) provides rich physical affordances (Norman, 1999) so that even children can immediately understand what they can do with this artifact without reading a manual.

TUI pursues these features further into the digital domain by giving physical form to digital information and computation, employing physical artifacts both as representations and controls for computational media. Its design challenge is a seamless extension of the physical affordances of the objects into the digital domain.

This chapter introduced the basic concept of TUI and a variety of examples of TUI applications to address the key properties of TUI and its design challenges. TUI is still in its infancy, and extensive research is required to identify the killer applications, scalableTUI toolkits, and a set of strong design principles.

The research of TUI which gives physical forms to digital information/computation naturally crosses with the paths of industrial/product design as well as environmental/architectural design. It has also made an impact on the media-arts/interactive-arts community. The author hopes that TUI design will contribute to promote those interdisciplinary design research initiatives in the HCI community to bring strong design culture as well as media-arts perspective to the scientific/academic world.

Mark Weiser's (1991) seminal paper on ubiquitous computing started with the following paragraph: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*"

I do believe that TUI is one of promising paths to his vision of invisible interface.

ACKNOWLEDGMENTS

The author would like to thank many of the former and current TMG students and colleagues in the MIT Media Lab for their contributions to the TUI research in the past 10 years at the MIT Media Lab. Especially thanks are due to Dr. John Underkoffler, Dr. Brygg Ullmer, Dr. Kimiko Ryokai, Dr. James Patten, Angela Chang, Hayes Raffle, Amanda Perks, and Oren Zuckerman. Thanks are also due to Things That Think and Digital Life Consortia at the Media Lab for their support of our research. The author also would like to thank Prof. Bill Buxton and Dr. George Fitzmaurice for our collaboration on the Bricks project in 1994 at the University of Toronto (Fitzmaurice, Ishii, & Buxton, 1995b) which laid the foundation of TUI, and Prof. Mitchel Resnick for his vision of Digital Manipulatives (Resnick et al., 1998) which influenced our design of educational TUI applications.

References

Aish, R. (1979). 3D input for CAAD systems. *Computer-Aided Design, 11*(2), 66–70.

Aish, R., & Noakes, P. (1984). Architecture without numbers—CAAD based on a 3D modelling system. *Computer-Aided Design, 16*(6), 321–328.

Anagnostou, G., Dewey, D., & Patera., A. (1989). Geometry-defining processors for engineering design and analysis. *The Visual Computer, 5*, 304–315.

Anderson, D., Frankel, J. L., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E., & Yedidia, J. S. (2000). Tangi-

ble interaction + graphical interpretation: a new approach to 3D modeling. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 393–402). ACM Press/Addison-Wesley Publishing Co.

Arias, E., Eden, H., & Fisher, G. (1997). Enhancing communication, facilitating shared understanding, and creating better artifacts by integrating physical and computational media for design. *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (pp. 1–12). ACM Press.

Brave, S., & Dahley, A. (1997). inTouch: A medium for haptic interpersonal communication. *Conference on Human Factors in Computing Systems* (*CHI '97*) Atlanta, GA (pp. 363–364). ACM.

Brave, S., Ishii, H., & Dahley, A. (1998). Tangible interfaces for remote collaboration and communication. *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (pp. 169–178). ACM Press.

Burbeck, S. (1992). Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC). http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

Buxton, W. (1995). Integrating the Periphery and Context: A New Model of Telematics. *Proceedings of Graphics Interface '95* (pp. 239–246).

Chang, A., O'Modhrain, S., Jacob, R., Gunther, E., & Ishii, H. (2002). ComTouch: design of a vibrotactile communication device. *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (pp. 312–320). ACM Press.

Chang, A., Resner, B., Koerner, B., Wang, X., & Ishii, H. (2001). LumiTouch: an emotional communication device. *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (pp. 313–314). ACM Press.

Cohen, J., Withgott, M., & Piernot, P. (1999). LogJam: a tangible multiperson interface for video logging. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the limit*, Pittsburgh, PA (pp. 128–135). ACM Press.

Crampton Smith, G. (1995). The Hand That Rocks the Cradle. *I.D.*, 60–65.

Dahley, A., Wisneski, C., & Ishii, H. (1998). Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space. *Conference on Human Factors in Computing Systems,* Los Angeles (pp. 269–270). ACM.

Fishkin, K. P. (2004). A taxonomy for and analysis of tangible interfaces. *Personal Ubiquitous Comput, 8*, 347–358.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. A. S. (1995a). Bricks: Laying the Foundations for Graspable User Interfaces. *Conference on Human Factors in Computing Systems,* Denver, Colorado (pp. 442–449). ACM.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. A. S. (1995b). Bricks: laying the foundations for graspable user interfaces, *Proceedings of the SIGCHI Conference on Human factors in computing systems* (pp. 442–449). ACM Press/Addison-Wesley Publishing Co.

Fogg, B., Cutler, L. D., Arnold, P., & Eisbach, C. (1998). HandJive: a device for interpersonal haptic entertainment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 57–64). ACM Press/Addison-Wesley Publishing Co.

Frazer, J. (1994). *An Evolutionary Architecture Architectural Association*, London.

Frazer, J., Frazer, J., & Frazer, P. (1980). Intelligent physical threedimensional modelling system. *Computer Graphics 80*, North Holland (pp. 359–370).

Frei, P., Su, V., Mikhak, B., & Ishii, H. (2000). curlybot: designing a new class of computational toys. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, the Netherlands (pp. 129–136). ACM Press.

Goldberg, A. (1984). *Smalltalk-80: The Interactive Programming Environment*. Addison-Wesley.

Gorbet, M., Orth, M., & Ishii, H. (1998). Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography. *Conference on Human Factors in Computing Systems* (*CHI '98*) (pp. 49–56). ACM.

Holmquist, L. E., Redstr, J., & Ljungstrand, P. (1999). Token-Based Acces to Digital Information. *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing* (pp. 234–245). Springer-Verlag.

Ishii, H. (2004). Bottles: A Transparent Interface as a Tribute to Mark Weiser. *IEICE Transactions on Information and Systems E87-D, 6*, 1299–1311.

Ishii, H., Fletcher, H. R., Lee, J., Choo, S., Berzowska, J., Wisneski, C., Cano, C., Hernandez, A., & Bulthaup, C. (1999). musicBottles. *ACM SIGGRAPH 99 Conference abstracts and applications* Los Angeles (p. 174). ACM Press.

Ishii, H., Mazalek, A., & Lee, J. (2001). Bottles as a minimal interface to access digital information. *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (pp. 187–188). ACM Press.

Ishii, H., Ratti, C., Piper, B., Wang, Y., Biderman, A., & Ben-Joseph, E. (2004). Bringing clay and sand into digital design—continuous tangible user interfaces. *BT Technology Journal, 22*(4), 287–299.

Ishii, H., & Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Conference on Human Factors in Computing Systems* (*CHI '97*), Atlanta, GA (pp. 234–241). ACM.

Ishii, H., Wisneski, C., Brave, S., Dahley, A., Gorbet, M., Ullmer, B., & Yarin, P. (1998). ambientROOM: Integrating Ambient Media with Architectural Space (video). *Conference on Human Factors in Computing Systems* (*CHI '98*) (pp. 173–174). ACM.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., & Mackey, K. (1989). The Xerox Star: a retrospective. *IEEE Computer, 22*(9), 11–26, 28–29.

Kitamura, Y., Itoh, Y., & Kishino, F. (2001). Real-time 3D interaction with ActiveCube. *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (pp. 355–356). ACM Press.

Klemmer, S. R., Thomsen, M., Phelps-Goodman, E., Lee, R., & Landay, J. A. (2002). Where do websites come from?: capturing and interacting with design history. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems: Changing our world, changing ourselves* (pp. 1–8). ACM Press.

Kobayashi, K., Hirano, M., Narita, A., & Ishii, H. (2003). A tangible interface for IP network simulation. *CHI '03 extended abstracts on Human Factors in Computing Systems* (pp. 800–801). ACM Press.

Lee, J., Su, V., Ren, S., & Ishii, H. (2000). HandSCAPE: a vectorizing tape measure for on-site measuring applications. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 137–144). ACM Press.

Mazalek, A., Wood, A., & Ishii, H. (2001). genieBottles: An Interactive Narrative in Bottles. *Conference Abstracts and Applications of SIGGRAPH '01* (pp. 189). ACM Press.

Norman, D. A. (1999). Affordance, conventions, and design. *Interactions*, pp. 38–43.

Pangaro, G., Maynes-Aminzade, D., & Ishii, H. (2002). The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. *Proceedings of the 15th annual ACM symposium on User Interface Software and Technology* (pp. 181–190). ACM Press.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Patten, J., Ishii, H., Hines, J., & Pangaro, G. (2001). Sensetable: a wireless object tracking platform for tangible user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 253–260). ACM Press.

Patten, J., Recht, B., & Ishii, H. (2002). Audiopad: A tag-based interface for musical performance. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression* (Dublin, Ireland, May

24–26, 2002). E. Brazil, Ed. New Interfaces for Musical Expression. National University of Singapore, Singapore, 1–6.

Piper, B., Ratti, C., & Ishii, H. (2002). Illuminating clay: a 3-D tangible interface for landscape analysis, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing our world, changing ourselves* (pp. 355–362). ACM Press.

Raffle, H. S., Parkes, A. J., & Ishii, H. (2004). Topobo: a constructive assembly system with kinetic memory. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 647–654). ACM Press.

Ratti, C., Wang, Y., Piper, B., Ishii, H., & Biderman, A. (2004). PHOXEL-SPACE: an interface for exploring volumetric data with physical voxels. *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (pp. 289–296). ACM Press.

Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhardt, U., & Meier, M. (1998). BUILD-IT: a planning tool for construction and design. *CHI '98 Conference Summary on Human Factors in Computing Systems* (pp. 177–178). ACM Press.

Rekimoto, J., Ullmer, B., & Oba, H. (2001). DataTiles: a modular platform for mixed physical and graphical interactions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 269–276). ACM Press.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998). Digital manipulatives: new toys to think with. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 281–287). ACM Press/Addison-Wesley Publishing Co.

Ryokai, K., Marti, S., & Ishii, H. (2004). I/O brush: drawing with everyday objects as ink. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 303–310). ACM Press.

Smith, D. (1982). Designing the Star User Interface. *Byte*, 242–282.

Stifelman, L. J. (1996). Augmenting real-world objects: a paper-based audio notebook. Conference companion on Human Factors in Computing Systems: Common ground (pp. 199–200). ACM Press.

Suzuki, H., & Kato, H. (1993). AlgoBlock: A tangible programming language—a tool for collaborative learning. *The 4th European Logo Conference* (pp. 297–303).

Ullmer, B., & Ishii, H. (1997). The metaDESK: Models and Prototypes for Tangible User Interfaces. *Symposium on User Interface Software and Technology* (*UIST '97*) (pp. 223-232). ACM Press.

Ullmer, B., & Ishii, H. (2000). Emerging frameworks for tangible user interfaces. *IBM Systems Journal, 39*(3–4), 915–931.

Ullmer, B., Ishii, H., & Glas, D. (1998). mediaBlocks: physical containers, transports, and controls for online media. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (pp. 379–386). ACM Press.

Ullmer, B., Ishii, H., & Jacob, R. J. K. (2003). Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries. INTERACT 2003 Conference, IFIP.

Ullmer, B., Ishii, H., & Jacob, R. J. K. (2005). *Token+constraint systems for tangible interaction with digital information, 12*, 81–118.

Underkoffler, J., & Ishii, H. (1998). Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface. *Conference on Human Factors in Computing Systems* (*CHI '98*) (pp. 542–549). ACM Press/Addison-Wesley Publishing Co.

Underkoffler, J., & Ishii, H. (1999). Urp: a luminous-tangible workbench for urban planning and design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the limit* (pp. 386–393). ACM Press.

Underkoffler, J., Ullmer, B., & Ishii, H. (1999). Emancipated pixels: real-world graphics in the luminous room. *Proceedings of the 26th annual Conference on Computer Graphics and Interactive Techniques* (pp. 385–392). ACM Press/Addison-Wesley Publishing Co.

Weiser, M. (1991). The computer for the 21st Century. *Scientific American, 265*(3), 94–104.

Wellner, P. (1993). Interacting with Paper on the DigitalDesk. *Communications of the ACM, 36*(7), 87–96.

Zuckerman, O., Arida, S., & Resnick, M. (2005). Extending tangible interfaces for education: digital montessori-inspired manipulatives. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 859–868). ACM Press.

Zuckerman, O., & Resnick, M. (2004). Hands-on modeling and simulation of systems. *Proceedings of the 2004 conference on Interaction Design and Children: Building a community* (pp. 157–158). ACM Press.

·25·

ACHIEVING PSYCHOLOGICAL SIMPLICITY: MEASURES AND METHODS TO REDUCE COGNITIVE COMPLEXITY

John C. Thomas and John T. Richards
IBM T. J. *Watson Research Center*

Scope and Structure of this Chapter **490**
Why Study Psychological Complexity? 490
The Nature of Psychological Complexity as a Variable 490
Complexity and Related Concepts **493**
Relationship of Complexity and Ease of Use 493
Relationship of Simplicity and Complexity 493
Relationship of Complexity and Uncertainty 494
Relationship of Complexity to Number 494
Relationship of Complexity to Nonlinearity 495
Relationship of Complexity to Distribution 495
Relationship of Complexity to Nature of Elements 495
Relationship of Complexity to Naming Scheme 495
Relationship of Complexity to Obscurity 496
Relationship of Complexity to Structural Framework . . . 496
Complexity of System vs. Task Complexity 496
Complexity of System vs. Contextual Complexity 497
Complexity, Feedback, and Interactivity 497
Four Sources of Difficulty for HCI **498**
Understanding the Syntax and Semantics of Communicating with the Computer 498
Making Tacit Knowledge Explicit 498
Making the Computation Efficient and Effective 498
Making the System Understandable and Maintainable 498
Sources of Complexity in the Development Process **499**
Radical Iteration in the Field 499
Problem Finding 499
Problem Formulation and Requirements 499
Design 500
Development 500
Testing 501
Deployment 501
Service 501
Maintenance 502
Ways to Measure Complexity **502**
A Priori Mathematical Models 502
Linear Regression 502
Subjective Measures 502
Textual Analysis of Documentation 503
Iterative Design and Testing 503
Possible Future Approaches **503**
Complexity Reduction in Practice: Case Studies **503**
Case Study 1: Query By Example 503
Case Study 2: Web Accessibility Technology 504
Conclusions **505**
References **505**

SCOPE AND STRUCTURE OF THIS CHAPTER

In this chapter, we explain why psychological complexity is (or should be) of interest to the designers of human computer systems. We then distinguish between intrinsic complexity and undue complexity. We presume that undue complexity is generally (but not universally) counterproductive in that it leads to more errors, frustration, and greater task completion times. Generally, these are all things to be avoided in a work-oriented context. However, in a more aesthetic, recreational, or pleasure-oriented context, increased psychological complexity can often be desirable. We examine the sources of undue complexity. We hypothesize undue complexity can arise intentionally, through incompetence, or (most commonly) as an unintended side effect of normal socialization processes.

We differentiate complexity from many related concepts such as uncertainty, obscurity, and difficulty. Then, we review various approaches to measuring complexity. We suggest where and how complexity may be introduced during the overall development of human-computer systems and several approaches that may help minimize undue complexity. Finally, we speculate on some possible future developments in the field of psychological complexity and briefly discuss two case studies.

Why Study Psychological Complexity?

Although mathematicians have treated complexity as a topic for a long time, more recently this interest has spread into many fields (e.g., Holland, 1995; Bar-Yam, 1997, 2000). The attempts to understand fields as diverse as economics, ecology, biology, and machine learning, among others, relate to similar mathematical treatments, often under the general rubric of Complex Adaptive Systems. Although there are still many unsolved problems in this field, one might raise the issue of why there needs to be a separate inquiry into the nature of psychological complexity. There are two basic reasons why psychological complexity deserves a separate treatment. First, what may be thought of as objectively complex may or may not be psychologically complex. Indeed, an exploration of these differences forms a major part of this chapter. Second, when we consider complexity in the more specific context of human-computer interaction (HCI), it is useful to differentiate intrinsic complexity from undue or gratuitous complexity. Some tasks are inherently complex. We may help people perform these tasks via work aids, education, documentation, rule-based systems, or the clever design of work groups, but some considerable intrinsic complexity may remain. In contrast, although regrettable, it still seems to be a fact that many systems, applications, and artifacts are unnecessarily complex. Poor design, for instance, burdens users with complexity beyond what is required by the nature of the task. In this chapter, we will explore sources of this undue complexity, as well as ways to prevent or mitigate it. While this chapter will reference some of the relevant literature on the more general topics of complexity and psychological complexity, the focus of the chapter is on how we can use these concepts to improve HCIs, in most cases by reducing undue complexity.

In terms of the underlying cause, there seem to be three main reasons for undue complexity. First, undue complexity sneaks into systems even though people consciously try to prevent it. This can happen for numerous reasons, which we will explore in detail below (e.g., lack of appropriate methodology). Second, seemingly undue complexity can be injected into systems intentionally. For instance, systems commonly enforce rules governing both password structure and password lifetime. These are often perceived as an annoyance by end users but are intentional policy decisions aimed at improving overall system security. A third and more subtle reason for undue complexity relates to ordinary social processes. As a group of people work together or live together, they develop and evolve ways of referring to things that become convenient short hands for the in-group but become increasingly obscure or difficult for the out-group. In the extreme, this results in different accents and dialects and eventually distinctly different natural languages, customs, and assumptions. Disparate groups of people end up, not simply using different terms, but thinking about the world differently (Ableу, 2005). This process is typically, though not exclusively, unconscious. This type of socialization begins early; experiments indicate that children as young as several weeks old are already less capable than they were at birth of making auditory distinctions within the phonemic categories deemed equivalent by their linguistic community. In one experiment, as children living in an English-speaking environment aged from 6 months to 12 months, their percentage of discriminations of a non-English distinction decreased from 80% correct to 10% (Werker & Tees, 1984). Categorical speech perception is only one dramatic form of a process that is happening all the time and at many levels in normal social interaction.

The Nature of Psychological Complexity as a Variable

Why are people interested in psychological complexity? It is an important intervening variable that simplifies analysis and prediction. On one hand, a number of factors, explored in detail in this chapter, impact psychological complexity, and on the other hand, a number of dependent variables depend on psychological complexity.

One might yet question the utility of complexity as a unifying variable. Although Card, Moran, and Newell (1983) talked about complex behavior (and the principle of Rationality to help explain how to build models of complex behaviors from simple ones), they do not treat complexity as a variable, per se. Despite this, under a wide variety of conditions, accurate predictions of complex human behavior are possible (see also, John, 1990; Gray, John, Stuart, Lawrence, & Atwood, 1990). However, modeling at this level of detail requires a substantial amount of work. The hope for complexity is that one might develop predictive measures of complexity that could be applied easily without special expertise that would still be highly predictive of human behavior. In the ideal case, for instance, a development team might calculate the complexity of several alternative design ideas for a human-computer system and make at least some of the many necessary design decisions without the need for assessing each one with costly behavioral observations.

We expect intuitively that increased psychological complexity is positively correlated with increased errors, increased time to complete a task, and decreased productivity, as well as increased frustration in a results-oriented work environment. (Even in such work contexts, there may be exceptions; e.g., when lengthy vigilance at an overly simple task may actually produce more errors than a somewhat more complex one.) Nonetheless, the overall presumption, both in the public mind and in the scientific literature is that unnecessary complexity is to be generally avoided on the grounds of reducing errors and improving productivity. The cost of complexity can be considerable. For example, half of the consumer products returned as malfunctioning are actually functioning as designed; they are just too complex for the customer to use (den Ouden, 2006). A well-meaning attempt to improve a homeland security system resulted in more errors, longer decision times, and a decreased probability that people would even use the system (Coskun & Grabowski, 2005). Although details vary, Fig. 25.1 shows the general relationship of complexity to performance and frustration.

However, the relationship of psychological complexity to more ludic (pleasure-oriented) and aesthetic variables is more complex. In general, the prevailing wisdom is that a moderate amount of complexity is aesthetically pleasing. Stimuli, responses, mappings between stimuli and responses or contexts that are either too complex or too simple are not as interesting, pleasurable, or engaging as those that are moderately so. This seems to be a robust phenomenon applying to infants as well as adult humans and to various species. Of course, it is still the case that what constitutes moderate complexity for an individual depends upon previous knowledge as well as personality predilections for more or less complexity. In any case, psychological complexity is potentially a powerful intervening variable that, on the one hand, promises to collapse the impact of numerous separate independent variables into one (e.g., complexity) and, on the other hand, to expand the implications of complexity onto a number of dependent variables. In the case of performance-related variables, the relationship is thought to be monotonic; in the case of subjective variables, the relationship is thought generally to be an inverted U-function, although it may vary according to the sense and the individual.

In the last few decades, most HCI work has examined work contexts in which productivity, in the broadest sense, has been

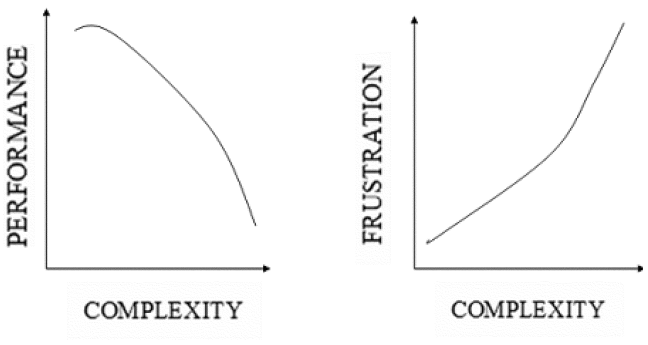


FIGURE 25.1. The general relationship between complexity and performance, and complexity and frustration.

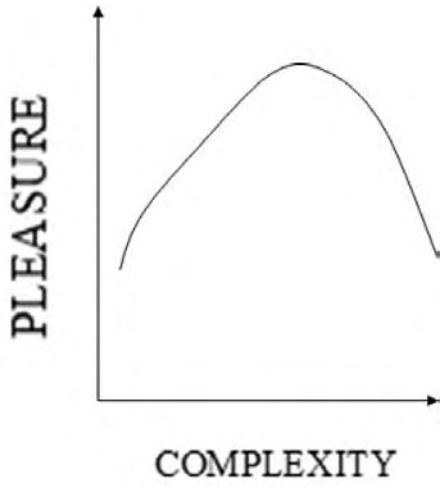


FIGURE 25.2. The typical relationship between complexity of a stimulus and the associated aesthetic pleasure in ludic situations.

a primary focus. As computing technology has become more intertwined with home life and entertainment, the importance of also using psychological complexity as a predictor of pleasure and preference is increasing.

In the domain of things that are meant to taste good, for instance, increasing complexity often seems to be presumed a good thing. For example, Vintage Cocktails (2007) created a subjective five-point scale for cocktails: simple, delicious but not daunting, sophisticated and smart, provocative and profound, and unusual and unforgettable. A blog of commentary on Macallan Elegancia scotch also indicates that complexity is presumed to be a good thing. For instance:

> The nose is very good balances, with pleasant sweet nutty and vanilla hints. The elegant balance does not mean any lack at complexity, with discrete wood and cut flower fragrances. In the palate, a clear nut taste with some slightly acid hints (citrus fruits) on a background of slightly bitter malt. A spicy and relatively long finish. (The Macallan: Speyside distillery (2007)

On the other hand, it is generally thought that visual and sonic patterns of moderate complexity are more often preferred. In the domains of art and architecture, some attempts have been made to quantify what makes for interesting or aesthetically pleasing patterns. For example, Christopher Alexander's (2002) work included an examination of and ordering of visual binary patterns, and Klinger and Salingaros (2000) suggested a metric for measuring the visual complexity of simple arrays. Other recent approaches attempt to relate preferences to fractal numbers (Taylor, Spehar, Wise, Clifford, Newell, Hagerhall, et al., 2005). Other investigators have tried to predict musical complexity based on objective factors (e.g., Shmulevich & Povel, 2000; Shmulevich, Yli-Harja, Coyle, Povel, & Lemstrom, 2001).

In more work-oriented contexts, psychological complexity is clearly related to other intervening variables such as stress, difficulty, and workload. However, there are distinctions among these. Stress depends not just on the complexity of the problem

situation itself but also on external factors such as the payoff matrix and internal factors like neuroticism. For instance, a specific puzzle might be thought to have a certain level of complexity. The amount of stress that one experiences in attempting to solve it, however, would depend upon the perceived rewards, punishments, and time pressure as well as the individual's internalized habits for viewing how various outcomes relate to overall life goals; for example, proclivities for awfulizing (Ellis, 2001) tend to increase stress far more than what might be objectively justified.

The more complex the task, the more difficult we would expect it to be; other things being equal. However, difficulty can also accrue to a task for a number of other reasons including environmental stressors such as extreme heat or cold, vibration, noise, or the necessity of applying large forces.

The term *workload* is more commonly applied to an individual or team across tasks. One can imagine an individual with a large number of tasks, each of which is simple, but whose job includes many interruptions and context switches from one simple task to another. In such a case, the workload may be high even though any particular task may be simple.

There is another sense in which the term *psychological complexity* is sometimes used. Social critics may use the term to refer not to a single task or system but to the totality of life experiences. Modern life in the information age is sometimes deemed too complex in its totality. This increased complexity may apply to politics, personal relationships, child rearing, transportation, healthcare, food choices, and so on, extending potentially to every aspect of modern life. Even providing for our entertainment needs has become a complex endeavor (Grinter & Edwards, 2005). In addition to the notion that each separate aspect of life is becoming more complex, people often find themselves multitasking and task switching (e.g., Gonzalez & Mark, 2005). While this is a potentially interesting avenue to explore in its own right, it is beyond the scope of this chapter, which instead will focus on psychological complexity as it applies to individual systems or tasks that have information technology aspects.

Early attempts to measure task difficulty with objective metrics include Attneave's (1957) figure complexity. There have been attempts to provide measures for the meaningfulness of trigram nonsense stimuli, responses, and the mapping between the stimulus and response in the verbal learning literature. Others have attempted to extend this work to real words; English words have been rated for frequency of occurrence (Kucera & Francis, 1967; Brown, G. D. A., 1984), concreteness (Paivio, Yuille, & Madigan, 1968), and so on. In one case, (e.g., Rubin, 1980), 51 dimensions of words were measured. Such measures do provide predictability in a wide range of laboratory tasks. For instance, the log of word frequency is closely related to decreased latency in a simple naming task where subjects are shown pictures and asked to name the pictured object as quickly as possible (Thomas, Fozard, & Waugh, 1977). Such empirical relationships as these, however, are problematic in terms of applicability to HCI. For one thing, when measures are taken on real words in English, many dimensions tend to be correlated. Carroll and White (1973) used a linear regression model and showed that age of acquisition is actually a more powerful predictor than frequency of occurrence (though the two are highly correlated). Neural net models of learning also demonstrate this effect (Smith, Cottrell, & Anderson, 2001). How might one apply such findings to HCI? For example, given a choice

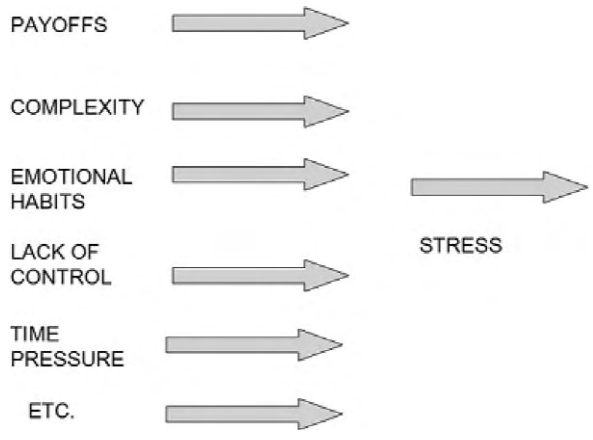


FIGURE 25.3. A number of variables, including complexity, can increase stress.

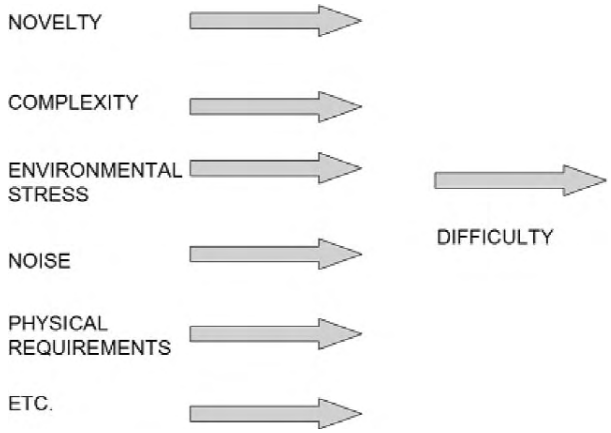


FIGURE 25.4. A number of variables, including complexity, can add to task difficulty.

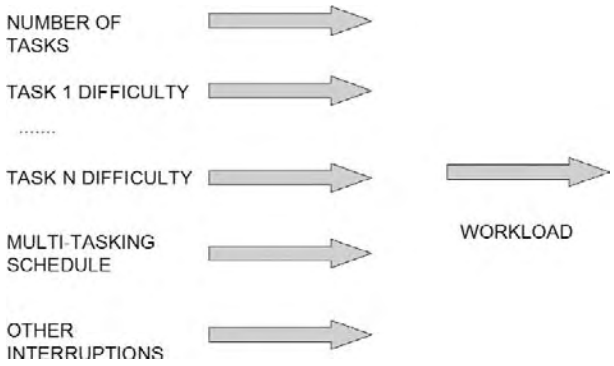


FIGURE 25.5. Overall workload is influenced by the difficulty of various tasks, the number of tasks, the scheduling of the tasks and other factors.

between two otherwise equally appropriate words, it is generally better to use the word that is more frequent in the language, or even better, one that is learned early in life. However, the relevance of this heuristic to the design of instructions, web pages, error messages, and so on, is very limited because in real design situations, we seldom have a choice between two otherwise equally appropriate words. More frequent words also often have the property of being more ambiguous, both semantically and syntactically (e.g., can, run), than longer, less frequent words. Words learned early in life often have little relevance to most tasks that adult users engage in. Knowledge of the particular users, their vocabularies, the task demands, conventions, contexts, and so on, dominates more general properties of the language in suggesting the best wording.

COMPLEXITY AND RELATED CONCEPTS

In this section, we will attempt to further define psychological complexity by distinguishing it from a number of similar concepts.

Relationship of Complexity and Ease of Use

It would seem that ease of use and simplicity constitute closely related concepts. However, they do differ in several ways. First, ease of use implies an empirical orientation. There are a number of interesting issues involved in measuring ease of use, such as choosing the sets of reference users, choosing representative tasks, and choosing the dependent variables to be measured (time to complete, errors, quality of result, etc.). In principle, ease of use can be measured objectively in terms of human behavior. Complexity, on the other hand, does not have the same degree of conceptual consensus. There are approaches that focus on the formal, intrinsic properties of a system or stimulus (e.g., Halstead, 1977; Alexander, 2002). Other approaches focus on the reactions of a human being to the system or stimulus. These can include passive measures, such as pupil dilation, staring time, or heart rate deceleration. They can also include more subjective judgments or attempts to measure various aspects of task-oriented behavior; for example, the number of steps in a process or the number of variables that must be remembered (e.g., Brown, A. B. & Hellerstein, 2004; Brown, A. B., Keller, & Hellerstein, 2005).

Nonetheless, other things being equal, one would expect something that is less complex to be easier to use. However, other things are typically not equal. Ease of use often depends heavily on people's expectations that may result from cultural conventions; often varying widely with time and place. Today, a new application that uses the typical GUI widgets, such as pull-down menus, may be easy to use for experienced users, primarily because people have a fair amount of positive transfer from earlier, similar experiences. A unique and unfamiliar but less complex interface may not prove as easy to use.

In addition, a design that maximizes visual simplicity through symmetry and minimalism may actually increase the difficulty of use. For example, a shower arrangement with one, clearly marked handle for hot water and another for cold water, combined with a two-position lever for shower and bath is more complex visually than a single, unlabeled knob with several degrees of freedom. At least initially, however, the former will be easier to use. Similarly, a normal bottle opener is not particularly elegant or simple in shape. However, its asymmetries, as well as its commonality, make it easy for people (in our culture) to use. An alternative bottle opener (See Fig. 25.6) consists of a metal cone with a thin protruding knob. This device is physically simpler, but harder to use, at least initially.

Relationship of Simplicity and Complexity

Simplicity and complexity seem to be opposites. However, there is an interesting asymmetry in their opposition. It is as though absolute simplicity is the center point of an *n*-dimensional sphere, while absolute complexity is the surface. In other words, there are many ways of moving away from simplicity toward complexity. Figure 25.7 illustrates just a few of the many ways that a system can become more complicated. For simplicity, we illustrate a two-dimensional rather than *n*-dimensional space.

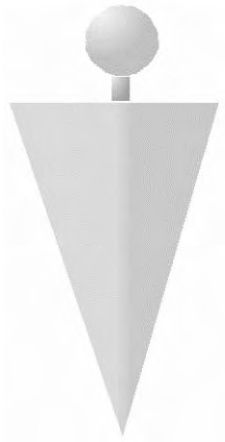


FIGURE 25.6. A "simple" but highly non-intuitive bottle-opener.

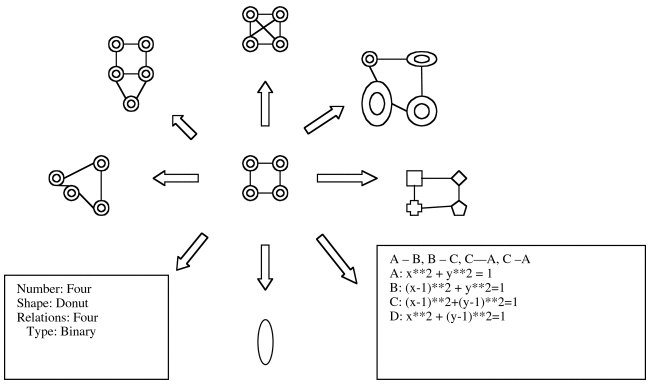


FIGURE 25.7. A "simple" system can be complicated in many different ways. (Here are just a few.)

This is more than an idle philosophical observation. In the real-world iterative cycle of design and behavioral observation, it is relatively difficult to move inward toward greater simplicity while improving ease of use. It is relatively easy to move along the surface of a sphere, resolving some types of complexity while simultaneously introducing others and therefore not increasing either overall simplicity or ease of use. For example, in designing a word-processing system, you might find that people are having difficulty finding desired items in long pull-down menus. You decide to make each menu simpler by changing the pull-down options depending on whether one clicks or clicks with the shift key depressed. This makes each menu simpler, but introduces another kind of complexity. Much of the art and the importance of experience in HCI can be conceived of as being able to move in from the surface rather than around the surface. (For a related discussion with respect to simplicity and accessability, see Lewis, in press.)

Relationship of Complexity and Uncertainty

Increased complexity is associated with increased uncertainty, *ceteris paribus.* The term *complexity* can be reasonably applied to stimuli, responses, or to the mapping between them. The concept of uncertainty however, seems to apply only to mapping. One can have uncertainty about what a stimulus is, or uncertainty about what response to make, but fundamentally, it has to do with what action is appropriate, given the current circumstances.

Complexity, as mentioned earlier, can be thought of either as something objectively measurable or as something subjective. The term *uncertainty* also has an objective meaning. If one turns over a randomly chosen card from an ordinary deck, there is more objective uncertainty about the outcome than if one flips a coin. The term *uncertainty* may also be applied in a purely subjective way. In this subjective sense, a given person may be completely certain that a randomly chosen card will be the Ace of Spades (because of nothing more than a compelling intuition) and completely uncertain about the outcome of a coin toss. The concept of subjective uncertainty has even further shades of application. One may watch the movie *Apollo 13* on numerous occasions. At one level, the outcome is already predefined and completely known. There is neither objective nor subjective uncertainty. Yet, within the inner context of watching the story play out, the viewers allow themselves (each time) to feel the subjective uncertainty of outcome.

Subjective uncertainty in HCI can apply at numerous levels. For example, in order to make touch typing a felicitous interaction, one subjectively presumes that each keystroke will be correctly transmitted to the computer and that what appears on the screen is an accurate mapping of what is actually being stored inside the computer. Objective certainty about the states of electromechanical systems probably never reaches 100%. However, as a strategy for partitioning effort, it is often useful to ignore very small error probabilities in subjective judgments. The first time the word-processing software changed Thomas' (this chapter) *the* to *The,* certainty changed. Thomas began to wonder what else the application might be doing. Later, a much worse problem surfaced in the form of an automatic update to styles. In response to something typed on page 1, something might change appearance on page 15. Subjective uncertainty also bears some resemblance to the concept of trust. You might have 1,000 interactions with someone, each of which provides evidence of trustworthiness, yet even a single interaction that shows a person to be untrustworthy may color a relationship for years to come. Similarly, if an interaction between user and the computer system violates an assumed behavioral norm by doing something completely unexpected, the level of subjective uncertainty created may be much greater than warranted in any objective sense. Moreover, if the user's experience includes such a violation, the scope of subjective uncertainty may be difficult to predict. One user may simply generalize uncertainty to a specific function; another may come to distrust a specific application (have a high degree of uncertainty) while a third may come to distrust computers in general.

Relationship of Complexity to Number

One would expect that if systems *A* and *B* are equivalent in other respects, but system *A* is more extensive than system *B* in terms of the number of items, that system *A* is more complex than system *B.* The degree to which an increase in number produces an increase in complexity, however, depends greatly on the structure of the systems. Suppose that systems *A* and *B* consist of 5 and 10 buttons respectively, combined with an OK light. The user has to use trial and error to discover the correct button to push to cause the OK light to light. On average, it will take 2.5 trials to find the right button for system *A* and 5 trials to find the right button for system *B*. Here, doubling the number of choices doubles the average number of trials. On the other hand, imagine that the user must find a sequence of 5 button pushes for system *A* and 10 button pushes for system *B.* When the user hits each correct button, the OK light comes on. On average, it will take the user $2.5 + 2 + 1.5 + 1 = 7$ attempts to find the right sequence for system *A* and $4.5 + 4 + 3.5 + 3 + 2.5 + 2 + 1.5 + 1 = 21$ attempts to find the right sequence for system *B.* Here a doubling in the number of buttons results in three times the mean number of trials. For a final case, imagine that the OK light comes on only after all buttons are pushed in the correct sequence. There are 5! possible sequences for system *A* and the user will find the correct sequence on average in 5!/2 trials or 60 trials. There are 10! possible sequences for system *B* and the user will find the correct sequence, on average in 1,814,400 trials. Here a doubling in number has led to an increase in trials by a factor of 30,240. In practical terms, system *A*, though cumbersome, could be solved by hand in a few minutes. System *B* would probably extend beyond anyone's patience. A quantitative difference in complexity actually would result in the qualitative difference between something doable and something not doable. These examples illustrate that the impact of number on complexity greatly depends on the nature of the system. In particular hidden states are usually a bad thing in terms of complexity, but just how bad they become rises quickly with increasing number.

Another example has to do with working memory and relates to George Miller's (1956) famous paper "The magic number seven plus or minus two." Though the size of working memory

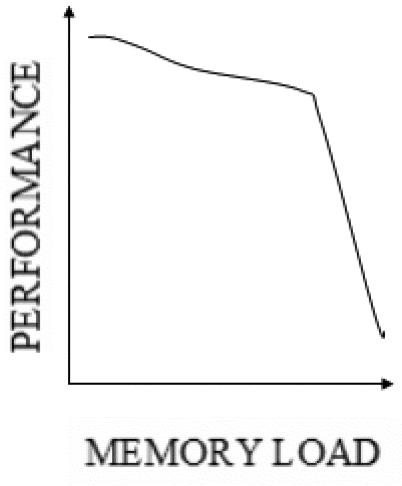


FIGURE 25.8. How memory load can have a minor effect on performance, up to a point, after which increasing memory load still more yields a sharp drop in performance.

varies somewhat depending on the number of dimensions and the type of material, it is not large. A system that allows a person to operate within that span can be nearly error free, while a system that forces a person to operate beyond that span will be extremely error prone. A modest increase in memory load can thus result in a very large increase in time and errors.

Relationship of Complexity to Nonlinearity

One of the defining characteristics of complex adaptive systems is that they exhibit nonlinearity. From the perspective of survival, this nonlinearity is a good thing. From the perspective of someone trying to interact with a computer system, nonlinearity can be a major source of difficulty in understanding, predicting, and controlling the behavior of the system. For example, Thomas (this chapter) once had to debug a PDP-8 program that was intended to collect reaction time data. There was only a single bit wrong in the entire program (which probably had roughly 5,000 bits total). In a linear system, you would expect that a .02% error would result in a small behavioral problem; for example, the reaction time would be off by a constant .02%. Of course, nothing of the kind happened. The program did not collect reaction times at all. Instead, it eventually caused the interpreter to crash. The bit that was wrongly set was the indirect bit so that instead of storing the reaction time, the program used the reaction time as an address in which to store the contents of a register. Furthermore, since every time the author tried to test the program, the reaction time varied slightly, the precise behavioral path of the program was different every single time. This example is not unusual, but it illustrates the property that computer systems are typically highly nonlinear in their behaviors. Generally, higher layers of software are designed to make the system, as seen by the user, as being more nearly linear (and therefore, more predictable). However, even in commonplace high-level applications, many nonlinear aspects typically remain.

Relationship of Complexity to Distribution

Halstead (1977) claimed that complexity C is equal to a constant k times the sum of the number of unique operators U times the natural log of the total number of operators u plus the number of unique operands O multiplied by the natural log of the total number of operands o.

$$C = k * (U + \ln u + O + \ln o)$$

This is not an unreasonable approximation. We have already seen how number can increase complexity. We could also consider, however, how these operators (or operands) are distributed throughout a program. Unlike Halstead (1977), are speculate that (a) AAAAAAAABBBBBBBBCCCCCCCC is easier to comprehend, use, edit, and so on, than (b) AABBCAABCBBCCAABCAACCCB.

It has been known for some time that repeated operations are typically faster than alternating ones (e.g., Fozard, Thomas, & Waugh, 1976). There can be exceptions to this general rule based on rhythm, fatigue, or satiation, especially for extremely simple tasks. However, for complex tasks, we generally expect a distribution such as (a) above to be much easier than (b). For example, in the Stroop task, one can be asked either to name the colors of ink (while ignoring the word) or to read the names of colors (while ignoring the color of ink). Alternating between the two every line is much slower than doing either task *en masse* (e.g., Philips, Bull, Adams, & Fraser, 2002). In the Brown et al. (2005) model, this source of complexity is modeled as context shifts.

Relationship of Complexity to Nature of Elements

Given the same number and structure of elements, two systems that are formally isomorphic may be quite different in terms of psychological complexity based on differences in human biology, learning, or both. For example, it seems clear that the human visual system is especially well tuned to handle the complexity of human faces. For example, Johnson and Morton (1991) presented evidence that babies as young as three hours after birth can recognize human faces. Chernoff (1973) suggested that this ability might be used to represent underlying complexity in a way that is easier for humans to deal with. In the auditory domain, it is clear that the difficulty of dealing with (recalling, rearranging, etc.) a string of phonemes that follow the phonological rules of one's native language is much less than dealing with an equal length string that violates those rules.

Relationship of Complexity to Naming Scheme

Consider two programs that manifest identical structures. The program asks the user to input an integer from 1 to 100. For each integer, the program then displays a corresponding unique horoscope. Structurally, this is a very simple program. However, imagine that in one version, the user's input number is called, "User's Input Number" and in another it is called "NXBNM." Further, suppose the horoscopes in the first program are labeled "horoscope1," "horoscope2," "horoscope3,"

and so on for each corresponding integer. In the second program, they are each labeled with a random two-letter string such as "HA," "IB," "JK," and so on. It is hopefully clear that these two programs, though formally identical would be vastly different in how easy they would be to understand, modify, debug, and so on.

While the above example is somewhat contrived, developing a consistent and intuitive naming scheme is a real-world problem. Often, when a programmer (or any other kind of designer) begins to name things, they do not have a complete understanding of the space of things they will have to name. For instance, when Thomas (this chapter) began working on a recent project, he named a file folder "ELFL" for "Electronic Learning Flow Language." As the project grew, one folder was much too generic for all the nuances that emerged. Furthermore, the focus of the project changed, and no one even talked about Electronic Learning Flow Language any more. Furnas, Gomez, Landauer, and Dumain (1982) found that not only are two different people likely to come up with different names for things, but that even the same person over time is unlikely to spontaneously come up with the same name. They suggest tables of synonyms to help alleviate this problem. Real computer systems are often rife with names that are difficult to comprehend or to recall precisely.

Relationship of Complexity to Obscurity

Consider the following two sequences of binary digits.

A: 0000 0001 0010 0011 0100 0101 0110 0111 1000
B: 1000 0101 0100 0001 0111 0110 0011 0010 0000

These two sequences each contain the same number of binary digits and the same number of groupings. The first sequence is essentially a counting sequence from zero to eight. This is an obvious sequence if one knows how to count in binary. The second sequence would presumably be much harder to memorize for most people. However, it also represents the numbers zero to eight; however, they are arranged in alphabetical order according to the English names. Once one realizes this rule, memorizing or reconstructing the second sequence becomes much easier. There is a sense in which the first sequence strikes us as a natural ordering because it is based on what we can see right before us. Numbers are likely to be highly associated with counting, after all. The rule that orders the second sequence strikes us as more obscure, however, since it depends on some other representation (the English names of the numbers) that is not naturally and strongly associated with the numbers *qua* numbers. This notion of obscurity is closely associated with the contrary concept of affordance (Norman, 1988). If the perceptible properties of an object or system immediately make it clear what it can do and how to do it we can say the object or system has good affordances. Of course, whether something appears obscure or not depends heavily on the user's previous experience, both cultural and personal and may include inborn factors as well. In Fig. 25.2, the equations version of the simple diagram in the middle illustrates obscurity.

Relationship of Complexity to Structural Framework

One of the interesting aspects of perceived complexity is that when it comes to human beings, sometimes more is less. For example, most English readers are capable of distinguishing *WORD* from *CORD* at a briefer presentation time than they can distinguish *W* from *C* (Reicher, 1969; Hildebrandt, Caplan, Sokol, & Torreano, 1995). Learning a set of paired associates, presented along with mnemonic visual images, is much easier than learning the set of paired associates alone (Thomas & Ruben, 1973). Subjects presented with a set of sentences and a context-setting theme or picture have a much easier time learning and remembering a story than those who do not have the picture (Bransford & Johnson, 1972). Recalling a story that fits with our cultural patterns is much easier than recalling one that does not (Bartlett, 1932). In all of the above cases, the advantage of the structural framework is that it relates new items to something that already exists in memory.

In other cases, however, the advantage of structure seems to have more to do with the underlying nature of the nervous system. For example, in attempts to develop useful and novel representations of speech signals in order to improve synthesis, one technique developed by Pickover (1985) mapped an autocorrelation function into polar coordinates scaled to fit within a 30 degree angle. The resulting dot patterns (see Figure 25.9) proved singularly difficult to interpret in any useful way. However, when the pattern was reflected to include a mirror image and the resulting 60 degree angle repeated six times around a central point to produce a snowflake-like pattern; the representation easily differentiated various vowel sounds as well as some subtleties like anticipatory nasalization which were virtually invisible in the traditional sonogram. Of interest in this context is the fact that the useful representation contained no more information than did the useless one. Arguably, one could say the speech-flake pattern was more complex. However, the complexity apparently allowed visual symmetry detectors to come into play and produced far more distinguishable and memorable patterns. These Symmetricized Dot Patterns have since been applied to numerous other domains as diverse as cardiac signals, mechanical stress, and even the crunchiness of cereals.

Complexity of System vs. Task Complexity

It is easy to fall into the trap of conflating the complexity of an interactive computer system with the complexity of the task that one is attempting to perform while using that system. In fact, many studies in the HCI literature compare two or more versions of systems by having subjects perform a small number of tasks in the two systems. Often these tasks are treated as fixed factors when they are actually a small sample (probably not randomly chosen) of all possible tasks. In principle, it is very difficult to generalize about the superiority of one system over another based on task performance. One cannot really know that a different set of tasks may not have completely obliterated or even reversed any observed trend.

As a general heuristic, one might suppose that the complexity of the system should reflect the complexity of the necessary tasks to be supported. That is, if the user only needs to

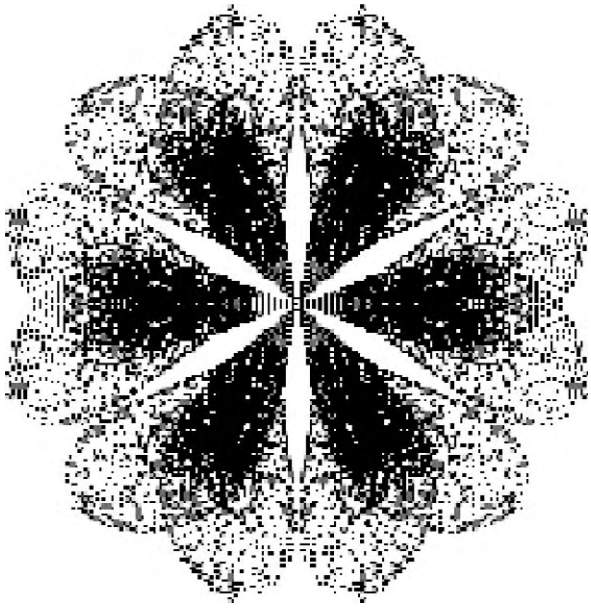


FIGURE 25.9. A symmetrized dot pattern.

perform very simple tasks, it may be enough to provide a simple system. However, if the user needs to perform very complex tasks, then a more complex tool may be required. In general, there may be some truth to this heuristic, but it must be applied with care. One caveat is that the function associated with this increased system complexity must actually be useful and allow the user to focus attention on the task. Otherwise, additional complexity in a system may actually be more disruptive when the task is complex than when it is simple. For example, consider the tasks of writing a two-page trip report and writing a full-length novel. Writing the novel (we assume) is a much more complex task. It may be that the various functions, fonts, formatting, and options in a complex word processor are unnecessary and even more distracting for someone trying to keep in mind a complex set of subplots and characters than for someone writing a short trip report.

Another important distinction is between the underlying complexity of a system and the complexity that is surfaced to the user. A huge amount of research and technological sophistication may be involved in the development of an automatic transmission but from the driver's perspective, the transmission is simple. From the perspective of the user, modern search engines provide a simple, commonplace interface. Most have no notion of the complexity of the underlying processes. Of course, attempts to do the user favors by providing hidden intelligence may sometimes backfire.

The ideal case is probably to provide a layered interface in which the system reveals only the complexity needed for the task. Early examples of this approach include the concept of "training wheels" (Carroll & Carrithers, 1984). The interfaces for another related set of projects, the Speech Filing System, the Audio Distribution System, and the Olympic Message System (Gould & Boies, 1984), also provided layered interfaces. However, in many real systems, it often happens that the user inadvertently falls through the user interface into a deeper layer or must do so to accomplish their real task. In addition, undue complexity in the underlying system can indirectly hurt the user experience by making code harder to test, debug, document, and maintain. Therefore, we consider undue complexity as something to be generally avoided both internally and in terms of what is to be made visible to the end user.

Complexity of System vs. Contextual Complexity

A person may be using a system, that in and of itself is simple, such as a bicycle, and their task may be simple, such as riding in a straight line without falling or crashing. However, riding that bicycle in a straight line on an empty, dedicated bike path through a park might be quite different from riding on a busy city street. It is somewhat of a judgment call to draw a line between task and context. In this example, one might just as well say that the two tasks are different. However, collapsing all such variations into the task complexity is probably counterproductive because the types of actions that can be taken and the power to take those actions are typically quite different in the case of simplifying the task versus simplifying the context.

Contextual complexity could increase overall complexity by providing secondary tasks or by bombarding the person with extraneous stimuli. Contextual complexity might also produce internal distracting material. Imagine two students using a word processor to complete an essay take-home exam. One of them is also worried about the outcome of some diagnostic medical tests. Consider two executives who are using presentation software to construct a sales pitch. One of them is operating in an organizational context that makes it necessary to pass the presentation through four layers of management, each with very different ideas about what makes a good sales presentation. It seems much more natural to consider such differences to be differences in contextual complexity rather than task complexity. We might expect the design of the presentation software to help the user deal with the task complexity in this example but not the contextual complexity.

Complexity, Feedback, and Interactivity

Finally, the extent to which we can easily understand, control, and predict the behavior of a system has much to do not only with how complex it is but also how readily we can interact with it and receive timely, unambiguous feedback. For example, the "dynamic query" system (Ahlberg, Williamson, & Shneiderman, 1992) does not decrease the complexity of the underlying data that one is attempting to understand, nor does it decrease the complexity of the interface by which one browses that data. In fact, the interface is actually more complex than a static browsing interface might be. However, it does allow fast, continuous, unambiguous feedback. Conversely, it is well known that delayed auditory feedback (or delayed visual feedback possible with television circuits) is highly disruptive to performance. In some settings, feedback on performance can still present complexity in the form of credit assignment. If you lose a game of chess, it is unambiguous that you lost, but determining what caused you to lose is difficult. This is not just a human problem; the "credit

assignment" issue is crucial to any complex adaptive system (Holland, 1995). In golf, putting can be a very difficult skill to improve. The main reason is that it is difficult to determine which of many possible factors is responsible for success or failure in putting in a real game situation. For example, if you miss a putt to the left it might mean that you misread the slope of the green; misread the grain of the green; hit the ball off center of the putter; hit the ball with a curved arc swing; hit the ball with a blade not normal to the path; and all combinations of these factors. In the domain of HCI (and elsewhere), this lack of unambiguous feedback about which subset of actions is causing which difficulties has sometimes been referred to as "tangled problems" (e.g., Carroll & Mack, 1984). In the domain of golf, Dave Pelz (2000), an MIT physicist and ex-astronaut, has invented a series of devices to give the learner unambiguous and differentiated feedback about these various possible sources of error. Studying these devices might be a valuable exercise for system designers hoping to provide a similar untangling for users.

FOUR SOURCES OF DIFFICULTY FOR HCI

One goal related to psychological complexity is to provide a system that will allow non-programmers to do what is essentially programming; that is, allow them to get a computer system to do what they want it to, not just in terms of choosing from preexisting options but to allow them to have more open-ended productive control. A whole succession of projects at Carnegie-Mellon University (e.g., Myers, McDaniel, Kosbie, & Marquise, 1993), the LOGO and related projects at MIT (e.g., Papert, 1993), and the long series of projects by Alan Kay and others, leading most recently to Squeak and Etoys (Kay, 2007), have attempted to reach this goal. Much of the focus of these projects was based on trying either to simplify the syntax and semantics of communicating with the computer or allowing it to take place in a manner more nearly like the way people communicate with each other. It is certainly true that the detailed and often obscure syntax of ordinary computer languages can provide a barrier to use. Weinberg (1971) pointed out, for example, that FORTRAN then had one set of rules for what constituted a valid arithmetic expression when used in an assignment statement and another, more restrictive set of rules for what constituted a valid arithmetic expression in an array index. Typically, programming languages are full of these kinds of details and no doubt, they provide unnecessary complexity for someone who does not spend a large amount of time using such a language. However, it is important to note that such details are only one of four potential sources of difficulty people may have in communicating their desires to a computing system. For concreteness, consider a scenario in which a chess grandmaster without programming experience desires to write a computer program that plays excellent chess.

Understanding the Syntax and Semantics of Communicating with the Computer

As stated, the chess grandmaster will probably have to learn the somewhat arbitrary syntax and semantics of a programming language. Although there is still promise in techniques aimed at programming by example and natural language communication, at this point, such systems are insufficient for writing something as complex as a chess program.

Making Tacit Knowledge Explicit

Assuming that the expert can master the rules of some programming language and its associated development environment, there are other difficulties. One of these is that the chess expert, like experts in many fields, is unlikely to be consciously aware of all the knowledge that he or she brings to bear on the game of chess. While some of this knowledge is somewhat explicit, such as develop the center, protect your king, and look for double attacks, much more of it exists in the form of patterns that have developed over the course of many experiences (deGroot, 1978; Simon & Barenfeld, 1969; Simon & Gilmartin, 1973). Making this tacit or implicit knowledge explicit, for fields of any depth and breadth, will probably prove a far more difficult and time-consuming task than learning the details of a specific programming language. The methods and techniques of knowledge programming and expert systems may be of some help here along with storytelling. Typically, one may elicit additional tacit knowledge (that is, transform it from tacit to explicit) by encouraging the expert to recall specific instances, asking them to generalize and then asking for counter-examples. It can also be useful to involve pairs or small groups in exchanging stories about their experiences.

Making the Computation Efficient and Effective

Another set of issues revolves around the efficiency of what is going on under the covers. Although computers are becoming ever more powerful and cheaper, it is still easy for nonexperts to write programs that are so inefficient as to be unworkable. A chess program that theoretically makes good moves but takes years for each move is unworkable and unusable. In fact, a naïve approach to writing a chess program is simply to do an exhaustive search to all finishes and work backwards. It may or may not be obvious to someone untrained in mathematics or computer science that such an approach would be completely unworkable.

While the chess case may be extreme, far less extreme cases can still be quite problematic. If one is writing an interactive program, timing issues are important and how to achieve good timing may require a great deal of expertise and knowledge. If systems involve multiple people and or computers, it is quite easy to introduce inconsistencies, deadlocks, thrashing, and so on, even if a nonprogrammer can master syntax and make implicit knowledge explicit. For a deeper examination of the relationship of usability and choices in the underlying architecture, see Bass and John (2003).

Making the System Understandable and Maintainable

In the ongoing stream of behavior, we make what seem to be obvious choices, such as where to put the car keys. If, a day later, we are prone to forget this obvious choice, imagine how much more difficult it can be to understand and recall the numerous

decisions that must be made in designing and implementing a computer system. A nonprofessional programmer may have very little knowledge of documentation, help systems, updates, security, compatibility issues, the process for reporting and fixing bugs, and so on. Consequently, even if the first three hurdles are overcome and a workable program is produced, its half-life may be very short indeed.

To summarize, the goal of having nonprogrammers directly instruct computers in a generative way has often focused on designing a very simple and consistent syntax for a programming language. In the above section, we indicated that such a focus only addresses the first of four major obstacles end-users face: to wit, learning a complex syntax.

While the possibility of programming by nonprogrammers is intriguing and probably quite a bit more difficult than it at first appears, there is a slight variant which is becoming more commonplace. There are instances where a community of practice includes some individuals who, although perhaps not professional programmers, are quite proficient at finding and modifying code in order to fulfill certain functions. In some cases, they may be using programming-like functions in an application program such as a spreadsheet. In other cases, they may actually be using full-fledged programming languages. By modifying existing code incrementally, these semi-expert users are often able to address all four of the issues mentioned above. An interesting early example of such a community was "Moose Crossing" (Bruckman, 1997), which provided a shared environment for kids to teach each other object-oriented programming in a special simplified language. Nonetheless, there remain many applications and systems that users interact with that are developed by professional programmers working in software companies. An examination of the way in which products are developed, in turn, can provide some insight into various places where undue complexity may be injected into systems (and therefore, how it might be reduced).

SOURCES OF COMPLEXITY IN THE DEVELOPMENT PROCESS

Radical Iteration in the Field

While this topic is dealt with in more detail elsewhere in this handbook (see also Greene, Jones, Matchen, & Thomas, 2003), it is worth at least a brief mention here that radical iteration in the field can greatly help avoid, prevent, and address all of the sources of complexity that we are about to enumerate. We will discuss two examples of this approach at the end of this chapter. By working as closely as possible with the potential users of a system doing their real work in their real context, one can avoid unnecessary complexity that might otherwise be injected during problem finding, problem formulation, design, development, deployment, and maintenance.

Problem Finding

Most of our typical education focuses on solving problems that other people have already found and formulated. Great leverage can arise from finding and formulating problems. Conversely, unnecessary complexity in a system can be injected from the very beginning by finding a problem that is not really a problem (for the users). A computer scientist, for instance, might discover that users intuitively choose a sequence of tasks to complete that is not quite as efficient as the theoretically optimal sequence. The users choose routes that are "good enough" and "satisficing" (Simon, 1962). However, the computer scientist (or accountant) might see the inefficiency as a problem that needs to be solved. The result may be a system that requires users to key in a long list of tasks to be done, resources to be used, the data required by the tasks and so on. The system that solves the (imaginary) problem could easily take far longer than the labor supposedly saved. Of course, there is the even more insidious problem that in order to model the efficiency of a process in the first place, certain simplifying assumptions must be made and these simplifications may well lead to a solution that is less nearly optimal in the real world than the original behavior. For example, in a large telecommunications company, a route optimization program was developed to dictate schedules to repair people. The program failed because it did not take into account many specific details that the repair people knew about the times and schedules and constraints of others with whom they had to coordinate.

Inadequate observation or starting with untested assumptions about what problems exist may also prevent people from even noticing easily solvable problems. On one factory tour in the early 1980s,Thomas (this chapter) was shown someone whose job was to precisely align two silver needles. The person was sitting in a very awkward position and behind the silver needles was a silver background. It was explained to me later that since the error rate was so high, they were working on a project to use machine vision to replace this position. The author pointed out that they might first try providing the person with ergonomic seating and a different background against which to align the needles. Automation proved unnecessary.

Problem Formulation and Requirements

One popular anecdote that illustrates the importance of proper problem formulation recounts a modern high-rise office building in which the office workers kept complaining about the slowness of the elevators. Computer programmers were called in and they reprogrammed the algorithms. The complaints increased. Engineers put in heavier duty cables and motors so that the elevators could move faster. Complaints remained as strong as ever, and several important multifloor tenants threatened to move out. In desperation, the building owner was considering sacrificing some of the floor space and adding additional elevator shafts at what would obviously be a high price. Someone suggested putting mirrors on all the floors near the elevators and the complaints ceased. In this case, people initially assumed that the problem to be solved was that the elevators ran too slowly and focused on various ways to increase the speed. Eventually, someone came along who realized that the real problem was that people were unhappy about the elevator speed. Mirrors gave them something to do and the time waiting did not seem so onerous.

In many real-world cases, this step is made more complex because typically there are a number of stakeholders, each of

which may have very different perceptions of the problems to be solved. A sociotechnical pattern, "Who Speaks for Wolf?" (Thomas, Danis, & Lee, 2002), based on a Native American story, transcribed by Paula Underwood (1991), suggested both the importance of finding all the relevant perspectives and stakeholders early in development and various techniques to try to accomplish that. Briefly, one of the members of a tribe was called "Wolf" because he made a life study of wolves. Once, while Wolf and a few other braves were on an extended hunting expedition, the tribe held council and decided they needed to move. A location was chosen, and the tribe moved; however, a few months later, it became obvious that the tribe had moved into midst of the spring breeding ground of the wolves. They had to decide whether to move again, post guards, or destroy the wolves. They finally decided to move again, but asked themselves, "What did we learn from this, and how can we avoid this kind of error in the future?" Someone pointed out that if Wolf had been present at the first council, he would have advised against the location. From then on, they decided that whenever they made a major decision, they would ask themselves, "Who speaks for Wolf?" to see whether there were missing stakeholders or perspectives that needed to be taken into account. Many projects could profit from such a process.

It is typically well understood that the earlier in the process of development an error is caught, the less expensive it is. An error in problem formulation can be extremely long-lived and expensive because it sets the context for measuring success. A product or system may be developed which appears to be successful at every step because the wrong thing is being measured so that ultimate failure goes unnoticed until too late. A classic example is the replacement of Coca-Cola with New Coke. According to Gladwell (2005), executives at The Coca-Cola Company were worried because they were losing some market share to Pepsi and blind taste tests indicated that Pepsi was more often preferred. As a result, a sweeter version of Coke that tasted more like Pepsi was developed and put on the market as a replacement for Coke. The reaction was surprising, immediate, passionate, and nearly disastrous for Coke. People wanted the old Coke back. The entire story probably involves brand loyalty, memory, and cognitive dissonance, but one fundamental problem was that the Coca-Cola executives were assuming that the goal was to develop a new product that was sweeter so that it would be preferred in taste tests over Pepsi. And, they succeeded—at solving the wrong problem. What tastes best when you take a decontextualized sip is not necessarily what you prefer (for a variety of reasons, not all directly related to taste) day after day, month after month. In contrast to the taste test, the case test shows what people actually buy over a long period of time, and of course, it is the latter measure which is actually important to profitability. In this case, the use of too simple a measure resulted in the wrong problem being solved. This is probably a common situation. Another example of this type occurred when the manufacturer for a new terminal for telephone operators did usability and productivity tests on operators using the terminal but failed to take into account the phone company customer who was also on the line and whose behavior actually turned out to be on the critical path most of the time. The entire design, development, testing, and so on was predicated on optimizing a system that consisted of the computer and the operator when the system that really mattered was the operator, the computer, and the customer. Again, an initial oversimplification resulted in a much more complex total solution than a more inclusive (and somewhat more complex) initial formulation would have produced.

However, undue complexity can also be introduced by formulating the problem in overly complex terms. The idea of (and failure of) detailed centralized economic planning may be the quintessential example.

Finally, undue complexity can be introduced by beginning with the wrong problem formulation. Even if a better formulation is discovered later, unless the development team is willing to throw out everything and start over, remnants of the original formulation will tend to persist into design, development, deployment, testing, and so on, making both the resulting system and its associated elements (e.g., sales and marketing materials, documentation, education packages, problem determination aids, etc.) more complex than they need have been.

Design

In an attempt to break down a complex problem into manageable subproblems, development teams, quite reasonably, divide into subteams to deal with various subsystems. Unfortunately, this can result in inconsistencies in basic functionality as well as in the user interface. For example, in one word processor, under certain conditions, the user would be faced with a message that said, in essence, "You cannot delete that file because it does not exist." An attempt to create another file by the same name however, resulted in a message that said, in essence, "You cannot create that file because it already exists." It seems clear that no commonly agreed upon definition of what it meant for a file to exist held sway through the whole of the development team.

At a more superficial level, one approach to reduce unnecessary discrepancies in the way that the user interface functions is to provide a style guide so that diverse developers or business partners working on various aspects or functions of an application suite will tend to provide a similar look and feel. There is much to be said for this approach if it is applied with perspective and intelligence. In the worst case, development teams may blindly follow what was meant as a guideline and interpret it as an ironclad rule. For instance, Thomas (this chapter) became involved in the development of an application for the service representatives for a large telecommunications company. The corporate development team insisted that we must follow some guidelines that claimed users should choose an object before choosing an action. For users who must move between multiple applications, there is a relative advantage of having consistency among applications. In this specific context, however, the users did not use multiple applications and it was clear that the way that they naturally thought of and interacted with the task, choosing an action first was far more intuitive, quicker, and less error prone than choosing an object first. Nonetheless, the management of the development team believed that the provided guidelines were received truth, and therefore they must be followed.

Development

Perhaps the greatest contributor to undue complexity is that the development team (and to a lesser extent, management, asso-

ciated marketing and sales, documentation specialists, etc.) becomes so familiar with a system that almost everything about it becomes obvious and easy, regardless of how it might be perceived by an end user. Once one sees a hidden figure (such as a pig in the clouds), it is nearly impossible not to see it. Over the lifecycle of a product, tens, hundreds, or even thousands of little conventions and assumptions become second nature to those associated with the development.

The cure for this malady is to continually test the scenarios of use, the designs, paper mock-ups, screen shots, prototypes, and beta versions with naïve users; that is, folks who are representative of potential users but not part of the design team. If, for any reason, this is not feasible (e.g., security, the users do not yet exist) other techniques can provide some amelioration. For instance, heuristic evaluation (Nielsen & Landauer, 1993) is likely to catch a fair proportion of actual problems. This is best done with HCI experts who are already familiar with both the technology and the application area. Typically, five to eight experts provide the most value. A variation on this technique involves having people successively take on different personae while interacting with a system and this may increase the number of errors found in the same period of time (DeSurvire & Thomas, 1993).

Testing

All too often, testing schedules become compressed. Partly as a result of time pressure and partly as a result of the fact that the development team has become accustomed to the high level and low level design, the error codes (or error messages) are often cryptic and designed for the productivity and convenience of the testing team. Unfortunately, in many cases, these error messages persist into the product that the ultimate end user experiences. Testers are often testing functions in a very well-specified context so that an error message is quite interpretable to them. However, to an end user, these same error messages are completely incomprehensible and often the output comes from dropping down several layers in the software stack. Thus, an end user attempting to hit a button in a high-level application may see an error message that not only does not specify what corrective action to take, but it mentions software elements that the end user is not even aware existed.

Another issue with testing, of course, is that testers who work in the development organization may only test reasonable combinations of function. Thomas (this chapter) designed the user experience for a "Dynamic Learning Environment" (Farrell, Thomas, Rubin, Gordin, Katriel, O'Donnell, et al., 2004). In testing the code, he tried altering the URL returned by the system in order to find what he was looking for. The developers all knew that you could not do this and it caused the system to crash—but it is an often-used strategy. For instance, if you know that http://www.umich.edu/~person1 is the URL of person 1, and you are trying to find the website of their colleague at Michigan, person 2, you might reasonably suspect that their URL might be http://www.umich.edu/~person2, and indeed this often works. Happily, he was able to convince the developers to prevent this ploy from crashing the system. More generally, it is important to have potential users test the system as well as those steeped in the cultural assumptions of IT generally and a particular company or product specifically.

Deployment

Today, many applications include "Wizards" to help unzip, install, and even use a product. Often, the distribution of applications is via websites, and therefore, finding the right application, finding the right version of the application, finding out whether one has all the prerequisites and necessary patches, and deciding which features and options to turn on or off can prove as complex, or even more complex, than actually using the application. Often, the instructions and interfaces associated with deployment suffer obscurities from the same root cause as those introduced by the development team—being overly familiar with the application or system. For example, in recently attempting to download an upgrade to a system, the instructions cautioned the user to be sure to use one specific URL and not another. This URL, however, did not refer to the actual URL for finding the desired download. The required URL laid three layers down a menu structure in a different application. To the folks who deployed this update, the context in which this specific URL was to be used was so obvious as to not bother making that context explicit.

Service

With the spread of an ever cheaper, higher bandwidth telecommunications infrastructure, service has experienced the dual trends of centralization of function and service being geographically distant from the users they are trying to support. While centralization offers some benefits in economy of scale and knowledge sharing, having service personnel geographically distant can mean a decrease in shared context between service people and the users they support. For example, local telephone operators used to be able to answer problems such as, "I need the number for that gas station across from the theater down town." Such problems are currently unanswerable from distant, centralized locations. In some cases, service personnel and users or customers may even have different native languages and come from different cultures. There is a possibility that some of this shared context may soon be reinstated, such as from geographical information systems and from systems that allow service people to view screens and actions remotely. It may be that a shared view of the real visual world is unnecessary; a shared representation of the salient features, assuming they can be identified, may be enough (O'Neill, Castellani, Grasso, Roulland, & Tolmie, 2005).

From a user's perspective, undue complexity is often injected into the process of joint problem solving with service personnel because the service personnel are often organized according to the underlying system that they are trying to support rather than the symptom experienced by a user. For instance, an end user may attempt to print something on a new printer and get an error message indicating that the print job failed. The user calls the help desk, and the top-level menu asks whether they are having a problem with connectivity, the operating system, or an application program, any of which may be the appropriate answer.

An apparently successful innovation in online help is the "Answer Garden" (Ackerman & Malone, 1990), which allows expert users over time with minimal disruption to grow a more detailed tree of FAQs. Other trends making the end user's job of getting help simpler include the use of search engines as well

as Wikis and weblogs to support communities of practice and communities of interest.

Maintenance

Consider the maintenance of an automobile circa 1950. While such machines were complex, the parts were all large enough to be visible, and when a part broke, it could be replaced by a similar part. Such repairs might be simple or they might be complex in that replacing a part might require moving other parts in order to reach the part to be replaced. Modern software systems (and such systems are becoming ubiquitously embedded in all other technologies) introduce a set of new and often complex maintenance issues. Before installing a new upgraded version of one piece of software, for instance, it is often necessary to check for several prerequisite pieces of software. Each of these may in turn require still other prerequisites. Even if you already have one of the prerequisites installed, say version *X* of program *Y*, you may also need to install a fair number of patches to *X.* In some cases, you need to uninstall software, taking care throughout to reboot as needed in order to clear out persistent memory. Indeed, if this entire tree is followed, the case still may be that installing some new piece of software causes something else not to work any longer. While the use of Wizards has made maintenance much simpler in many cases, if something does fail, diagnosing and fixing the failure may require delving into several layers of software beneath the GUI that is supposed to provide the simple interface for the end user.

WAYS TO MEASURE COMPLEXITY

A Priori Mathematical Models

We have already mentioned an attempt to measure the complexity of programs by Halstead (1977). Other measures have been proposed; probably the best known are those by McCabe and Butler (1986; McCabe, 1976) and Jones (1996). These metrics aim to capture the inherent psychological complexity of the program and not just the behavior that the program surfaces to the end user. We might think to find useful analogues here to measures for the complexity of the user's interaction with a program as well. Unfortunately, the measures do not currently deal with several major factors of psychological complexity. First, the models do not deal with the complexity of the visual (or other sensory) stimuli presented to the user. Second, the models do not deal with the complexity of the response required of the user. Third, the models do not take into account the many complicating factors mentioned above that impact the correspondence between what is required of the user's behavior and what the user already knows.

One of the first attempts to quantitatively predict ease of use *a priori* tried to measure the complexity of the internal structure necessary to generate behavior (Reisner, 1984). A similar approach has been used to try to predict learning time (Bovair, Kieras, & Polson, 1990). Basically, both approaches consider the complexity of the underlying rule systems. Reisner's claim is that, other things being equal, it takes longer to execute a more complex rule set. Bovair, Kieras, and Polson claimed that it takes longer to learn a more complex rule set as well as to execute it once learned. Both of these claims seem justified, though only with the caution of *ceteris paribus* (see Rautenberg, 1996 for a meta-analysis of quantification attempts of user interfaces; see Ivory & Hearst, 2001 for a broader review of attempts to automate all or part of the process of usability evaluation).

Linear Regression

Psychological tests have often been constructed by taking a very large number of diverse items and then seeing which ones correlate with a desired criterion. Of course, one needs to revalidate these items with a new sample, but it is interesting that somewhat predictive tests have been constructed with this method for many domains. In fact, linear predictive models have been successfully applied to a large number of domains and are generally better than human experts (see Dawes, 1982; Dawes, Faust, & Meehl, 1992). Walston & Felix (1977) used such a technique to predict the effort needed to complete a wide selection of software development projects. Interestingly, the most important factors had to do with the sociopolitical aspects of projects rather than the technical ones. A widely used, flexible method to predict various aspects of software development efforts is COCOMO (Boehm, 1981; 2000). Though not strictly linear, the calculations are based on an empirical analysis of actual software development projects.

Perhaps such approaches can be profitably extended to psychological complexity as well. In fact, at least one attempt to measure the "goodness" (related at least in part to complexity) of websites used a similar approach (Ivory & Hearst, 2001) and showed some evidence of success. However, in their study, somewhat different factors were predictive for different topic areas. Further, because conventions and underlying technology keep changing, it is likely that linear predictive equations might have to be updated on a frequent basis. In general, this is a potential limitation of linear predictive models. They can be quite accurate and quite robust, but only provided the underlying conditions remain relatively constant. The absence of a theoretical underpinning can lead to predictions that can become inaccurate without warning.

Subjective Measures

An entirely different approach to measuring cognitive complexity is to use subjective measures; in effect, to ask people to rate or rank items in terms of cognitive complexity. In judgments of simple stimulus sequences, it appears that complexity and randomness are highly related (Falk & Konold, 1997). An interesting and sophisticated attempt to provide a more differentiated subjective view of complexity applicable to real-world systems is that of Cognitive Dimensions of Complexity (Green, 1989; Green & Petre, 1996). This approach identifies 14 different dimensions of complexity. It can be used as a practical tool to focus the attention of developers successively on various areas of design that can lead to undue complexity. More recently, related work has focused on mismatches between the concepts inherent in the design of an artifact and the way users conceptualize (Connell, Blandford, & Green, 2004).

Textual Analysis of Documentation

In general, we expect the complexity of a description of something to correlate highly with the complexity of that thing. Of course, it is possible to describe a simple thing with undue complexity, but in most circumstances, both extrinsic reward structures and the rules of conversation tend to push description to be sufficiently complex to describe something accurately but sufficiently simple to be understood. Therefore, one promising ersatz measure of the complexity of something, such as a computer system or a procedure using that system, is to measure the complexity of the description. One commonly used measure is the Flesch Formula (Flesch, 1948), which basically uses the average length of words in letters and the length of sentences in words in a document to provide as a score the educational reading level required to understand that document. Another metric is the Gunning Fog index (Gunning, 1952), which seeks to measure the clarity of writing. A still more sophisticated automated approach is possible with the DICTION program (Hart, 2001) which gives 31 primary and several derived dimensions to describe the rhetorical style of a document. One of the primary dimensions is labeled complexity and several of the others might relate. Although this approach offers promise as a metric to be applied after the fact, caution must be applied to using it as an in-process metric. Otherwise, writers could use various tricks to make the description of an artifact appear simple while in reality making it incomprehensible. (In the absurd limiting case, one could provide a manual for a complex product that simply said, "Use it.") Clearly, documents must also be constrained to be complete in order to apply a fair and useful application of complexity metrics to descriptions.

Iterative Design and Testing

Clearly, much of the rest of this handbook expands on this topic. While considerations of complexity may prove useful, at this point, there is no substitute for interacting with real users doing real work in their real context as a general method for designing, developing, and deploying truly useful and usable systems. Thinking about and measuring complexity may well help in this process, but it will probably never substitute for it.

POSSIBLE FUTURE APPROACHES

One can imagine that ever more sophisticated techniques for brain activity imaging and other physiological measures may someday render a reliable and easy-to-use objective index of psychological complexity. Somewhat more likely is the continued evolution of modeling approaches, such as SOAR (Laird, Newell, & Rosenbloom, 1987) and EPIC (Kieras & Meyer, 1997). The follow-on programs might someday conceivably and automatically calculate a set of complexity numbers for combinations of user, task, system, and context. In the nearer term, it seems likely that continued improvement of heuristic approaches that do require some human intervention will prove useful (e.g., Dimensions of Cognitive Complexity; Green, 1989) and the Brown, Keller, and Hellerstein (2005) model in helping to design better user experiences. The latter model assumes that an expert path is known and complexity depends upon the number of actions, the memory load associated with retaining needed parameters, and the number of context shifts.

In our own work, we are attempting to extend this model in order to account more completely for points of uncertainty and the difficulties of decision making. In addition, we are building tools that enable complexity metrics to be produced as a side effect of normal development processes. In this way, developers may gain feedback that is both timelier and more differentiated with respect to their design decisions. In the short term, such metrics should provide useful feedback. As the tools come to be used over time, we expect the development teams to internalize the implications so that better initial design decisions are made.

COMPLEXITY REDUCTION IN PRACTICE: CASE STUDIES

Clearly, complexity is itself a complex topic. However, it is not intractable. Numerous examples of good design and development leading to suitable levels of complexity may be found. We conclude this chapter with two case studies from our own laboratory that illustrate how complexity may be considerably reduced in a real world setting.

Case Study 1: Query By Example

Query By Example, invented by Moshe Zloof (1975) in the early 1970s, was an attempt to provide a query language easy enough for computer-naïve novices to learn to use. Early studies indicated that nonprogramming high-school and college students were fairly successful in translating English questions into the syntax of this formal query language (Thomas & Gould, 1975). One probable reason for the relatively fast times and low error rates was that, in many cases, users only had to select the right place to write information rather than having to write all information from scratch (see Fig. 25.10). Another likely reason was that, in Halstead's (1977) terms, Query By Example was quite "dense." That is, almost all of the symbols required for a given query were symbols that would be required in any conceivable query language. There was little overhead. In other words, the intrinsic complexity of any given query remained, but there was no gratuitous or undue complexity introduced by the system itself.

Employees

Name	Age	Salary	Year of Hire
p.	>65		

"Print the names of everyone who is more than 65 years of age." Only the "p." and the ">65" is entered by the user.

FIGURE 25.10. A simple query in "Query By Example."

In some cases, students continued to have some difficulties, but further studies showed that these difficulties did not spring from the query language per se, but from general difficulties with logic (Thomas, 1976). For example, students had some difficulties with quantifiers (*all* and *some*), logical connectives (*and, or,* and *not*) and with operational distinctions (*sum, count*, and *count unique*). These same difficulties occurred in the absence of any requirements of the query language. Additional difficulties were encountered when the structure of the database did not match well with the structure of the question. For instance, if the database contained a column labeled "Year of Hire," a question asking for the names of people "hired after 1970" was easy. A question asking for the names of people "who have worked for more than 35 years," however, was much more problematic.

In a follow-on study (Thomas, 1983), students were not given English questions to translate but instead were shown a data base structure and given problem statements. They were then to generate their own English questions relevant to the problem and then translate their own questions into Query By Example. These results were much less encouraging. These students had very little concept of what types of questions were reasonable to put to the computer. One problem given was, "Some of the younger faculty members feel that they are not paid enough relative to the older faculty. Write a question that you think might shed light on this issue." A typical question was, "Are the younger faculty paid enough?" In other words, many of the students expected the computer to simply "formulate and solve the problem" for them.

In the 1970s, novice computer users not only might have trouble writing a correct query; they might also write an incorrect query that would consume a huge amount of resource. For this reason, it was considered worthwhile to try to predict the chances that a query was correct before submitting it to the search engine. Based on the formal properties of the query alone (number of operators, etc.), we were able to predict 54% of the variance in query accuracy. If we added the time taken to write the query and the student's own confidence rating, we achieved an accuracy of 75%. Finally, if we added a term for whether the mapping between the English and query was straightforward, more than 90% of the variance was predictable. All these measures may be thought of as factors related to psychological complexity. While the interface and language were developed and improved iteratively, our studies also showed that remaining user difficulties were due to intrinsic task complexity.

Case Study 2: Web Accessibility Technology

With the web growing in importance to both recreation and work, it became clear to us some time ago that barriers to web use needed to be lowered. It was no longer acceptable for people with visual and motor disabilities to be denied effective web access.

We began our explorations in this area by working directly with older adults, a population needing both web content modifications (text and image enlargement, navigational simplification, visual contrast modifications, etc.) and input adaptations (to filter out hand tremors, remove extraneous key presses and mouse clicks, etc.). Of course, modern browsers and operating systems provide a number of these modifications and adaptations, but access to them is distributed throughout the system, inconsistently surfaced, and often inaccessible to those most in need. Our challenge lay in finding a way to reduce the complexity of so much functionality simultaneously while adding still more accessibility features not otherwise available.

Our approach was iterative and user centered. Early on, it became clear that our users did not want a simplified browser. Nor did they want to be restricted to a subset of the web. After multiple attempts (Richards and Hanson, 2004; Hanson and Richards, 2005), we converged on a design that added a set of very simple control panels to the bottom of an otherwise normal browser, each control panel being dedicated to one class of adaptation or transformation. Figure 25.11 shows one such panel in the context of the full browser.

The only change to the browser's normal interface is the addition of a single "Settings" button. Clicking on this button brings up the first of the series of control panels. Clicking on one of the buttons in a control panel applies an immediate change to the web content. If the change is desired it is kept by doing nothing more with the panel. Otherwise, it is cancelled by clicking on a "None" or "Standard" button depending on the panel. The panels are organized in a conceptual ring. Clicking on the ">" button brings up the next panel. Clicking on the "<" button brings up the previous panel. Clicking on the "?" button brings up an interactive help page keyed to the current panel, which allows the user to easily explore the full range of modification the panel exposes.

A number of mechanisms lie beneath this simple surface. Some modifications are applied by altering browser settings stored in the system's global registry. Others are applied by automatically creating and installing a user style sheet. Others are applied by adjusting parameters in the underlying accessibility settings of the system. Others are applied by modifying the web page's in-memory Document Object Model. Still others are applied by means of always-running agents examining input event streams. All of this complexity is hidden from the user.



FIGURE 25.11. The "colors" panel for web accessibility software.

For a concrete example of the resulting simplification, consider the case of changing the colors of web pages. This is a modification often found useful for people with visual disabilities and for people with various forms of dyslexia. To change just the text foreground and background colors on a popular browser requires 16 separate steps. To ensure that link, visited link, and hover colors are suitably contrasting requires 24 additional steps. In contrast, in our design, setting two of the most popular color combinations—black text on white backgrounds or white text on black backgrounds—requires only a single click on a button showing the desired color combination. Optimally contrasting link, visited link, and hover colors are automatically set as a side effect of this choice. For those needing a broader range of choice, another simple panel allows the direct setting of the foreground and background RGB values. Again, the link, visited link, and hover colors are set automatically.

This case study illustrates a number of approaches to reducing software complexity. First, iterative user-centered design allowed for the rapid exploration of a range of alternative interfaces. Second, the addition of a new interface layer allowed for the unification of a number of disparate mechanisms. Third, careful analysis of the needs of users allowed for some choices to be surfaced prominently and others to be removed. Fourth, the provision of immediate feedback allowed users to directly experience the effects of any choice, trying them alone or in combination until their overall web experience was optimal.

CONCLUSIONS

Not surprisingly, psychological complexity is itself a complex topic. It should be of interest to designers of HCI because they typically need to reduce undue complexity. There are occasional exceptions, for instance, in entertainment and learning applications where the designer may want to increase psychological complexity. In this chapter, we have distinguished complexity from many related but different concepts such as uncertainty. Also, we have explored how undue complexity may be introduced into the development process and have suggested approaches to measure and reduce undue complexity. As the intrinsic complexity of the world seems to be on the increase, the importance of reducing undue complexity will continue to increase.

References

Abley, M. (2005), *Spoken here: Travels among threatened languages.* London: Arrow Books.

Ackerman, M., & Malone, T. (1990, April). Answer garden: A tool for growing organizational memory. *Proceedings of the ACM Conference on Office Information Systems*, Cambridge, MA, 31–39.

Ahlberg, C., Williamson, C., & Shneiderman, B. (1992). Dynamic queries for information exploration: An implementation and evaluation, *Proceedings of the ACM CHI'92: Human Factors in Computing Systems* (pp. 619–626). New York: ACM.

Alexander, C. A. (2002). *The nature of order: An essay on the art of building and the nature of the universe. Book One: The phenomenon of life.* Berkeley, CA: Center for Environmental Structure.

Attneave, F. (1957). Physical determinants of the judged complexity of shape. *Journal of Experimental Psychology, 53*, 221–227.

Barlett, F. C. (1932). *Remembering: An experimental and social study.* Cambridge: Cambridge University Press.

Bar-Yam, Y. (1997). *Dynamics of complex systems* (*Studies in nonlinearity*). Boulder, CO: Westview Press.

Bar-Yam, Y. (2000). Unifying themes in complex systems. *Proceedings of the international conference on complex systems.* New York, NY: Westview Press.

Bass, L., & John, B. E. (2003). Linking usability to software architecture patterns through general scenarios. *The Journal of Systems and Software, 66*, 187–197.

Boehm, B. W. (1981). *Software engineering economics.* Upper Saddle River, New Jersey: Prentice Hall.

Boehm, B. W. (2000). *Software cost estimation with COCOMO 2000.* Upper Saddle River, New Jersey: Prentice Hall.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*(1), 1–48.

Bransford, J., & Franks, J. (1972). Contextual prerequisites for understanding: Some investigations of comprehension and recall. *Journal of Verbal Learning and Verbal Behavior, 11*, 717–726.

Brown, A. B., & Hellerstein, J. L. (2004, September). An approach to benchmarking configuration complexity. *Proceedings of the 11th ACM SIGOPS European Workshop.*

Brown, A. B., Keller, A., & Hellerstein, J. L. (2005, May). A model of configuration complexity and its application to a change management system. *Proceedings of the Ninth IFIP/IEE International Symposium on Integrated Network Management* (*IM 2005*) (pp. 631–644). Nice, FR.

Brown, G. D. A. (1984). A frequency count of 190,000 words in the London-Lund Corpus of English Conversation. *Behavioral Research Methods Instrumentation and Computers, 16*, 502–532.

Bruckman, A. S. (1997). Moose crossing: Construction, community, and learning in a networked virtual world for kids. *Dissertation Abstracts International.* (UMI No. AAI0598541).

Carroll, J., & Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM, 27*(8), 800–806.

Carroll, J., & Mack, R. L. (1984). Learning to use a word processor by doing, by thinking, and by knowing. In J. Thomas & M. Schneider (Eds.), *Human Factors in Computer Systems* (pp. 13–51). Norwood, New Jersey: Ablex.

Carroll, J., & White, M. (1973). Age-of-acquisition norms for 220 picturable nouns. *Journal of Verbal Learning and Verbal Behavior, 12*(5), 563–576.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Society, 68*, 361–368.

Connell, I., Blandford, A., & Green, T. G. R. (2004). CASSM and cognitive walkthrough: Usability issues with ticket vending machines. *Behavior and Information Technology, 23*(5), 307–320.

Coskun, E., & Grabowski, M. (2005). Impacts of user interface complexity on user acceptance and performance in safety-critical systems. *Emergency Management, 2*(1), 1–29.

Dawes, R. (1982). The robust beauty of improper linear models in decision-making. In D. Kahneman, P. Slovic, & A. Tversky (Eds.), *Judgment under uncertainty: Heuristics and biases* (pp. 391–407). Cambridge: Cambridge University Press.

Dawes, R., Faust, D., & Meehl, P. (1989). Clinical versus actuarial judgment. *Science, 241*, 1668–1674.

DeGroot, A. D. (1978). *Thought and choice in chess* (*2nd edition*). The Hague, Netherlands: Mouton.

DeSurvire, H., & Thomas, J. C. (1993). Enhancing the performance of interface evaluation using non-empirical usability methods. *Proceedings of the 37th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 1132–1136). Santa Monica, CA.

Ellis, A. (2001). *Overcoming destructive beliefs, feelings and behaviors: New directions in rational emotive behavior therapy*. New York: Albert Ellis Institute.

Falk, R., & Konold, C. (1997). Making sense of randomness: Implicit encoding as a basis of judgment. *Psychological Review, 104*, 301–318.

Farrell, R., Thomas, J., Rubin, B., Gordin, D., Katriel, A., O'Donnell, et al. (2004, November). Personalized just-in-time dynamic assembly of learning objects. *E-learning 2004* (pp. 607–614).

Flesch, R. F. (1948). A new readability yardstick. *Journal of Applied Psychology,32*, 221–233.

Fozard, J. L., Thomas, J. C., & Waugh, N. C. (1976). Effects of age and frequency of stimulus repetitions on two-choice reaction time. *Journal of Gerontology*, *31*(5), 556–563.

Furnas, G. W., Gomez, L. M., Landauer, T. K., & Dumais, S. (1982, March). Statistical semantics: How can a computer use what people name things to guess what things people mean when they name things? In J. Nichols & M. Schneider (Ed.), *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 251–253). Gaithersburg, MD: ACM.

Gladwell, M. (2005). *Blink: The power of thinking without thinking.* New York: Little Brown.

Gonzalez, V. M., & Mark, G. (2005). Managing currents of work: Multitasking among multiple collaborations. *ECSCW 2005: Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work* (pp. 143–162). Netherlands: Springer.

Gould, J. D., & Boies, S. J. (1984). Speech filing: An office system for principals. *IBM Systems Journal, 23*(1), 65–81.

Gray, W. D., John, B. E., Stuart, R., Lawrence, D., & Atwood, M. E. (1990). GOMS meets the phone company: Analytic modeling applied to real-world problems. *Proceedings of IFIP Interact '90: Human Computer Interaction* (pp. 29–34). Cambridge, UK.

Greene, S. L., Jones, L., Matchen, P., & Thomas, J. C. (2003). Iterative development in the field. *IBM Systems Journal, 42*(4), 594–612.

Green, T. G. R. (1989). Cognitive dimensions of notations. In A. Sutcliffe & L. Macaulay (Eds.), *People and computers V* (pp. 443–460). Cambridge: Cambridge University Press.

Green, T. G. R., & Petre, M. (1996). Usability analysis of visual programming environments: A "cognitive dimensions" approach. *Journal of visual languages and computing, 7*, 131–174.

Grinter, R. E., & Edwards, W. K. (2005). The work to make a home network work. *ECSCW 2005: Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work* (pp. 469–488). Netherlands: Springer.

Gunning, R. (1952). *The Technique of Clear Writing.* New York: McGraw-Hill International Book Co.

Halstead, M. H. (1977). *Elements of software science* (Operating and programming systems series). New York: Elsevier.

Hanson, V. L., & Richards, J. T. (2005, May–June). Achieving a more usable World Wide web. *Behaviour & Information Technology, 24*(3), 231–246.

Hart, R. P. (2001). Redeveloping diction: Theoretical considerations. In M. West (Ed.), *Theory, method and practice in computer content analysis* (pp. 43–60). Westport, CT: Ablex.

Hildebrandt, N., Caplan, D., Sokol, S., & Torreano, L. (1995, January). Lexical factors in the word-superiority effect. *Memory and Cognition, 23*(1), 23–33.

Holland, J. J. (1995). *Hidden order: How adaptation builds complexity.* New York: Basic Books.

Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys, 33*(4), 470–516.

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. *Proceedings of CHI'90* (pp. 107–115). New York.

Johnson, M. H., & Morton, J. (1991). *Biology and cognitive development: The case of face recognition.* Cambridge, MA: Blackwell.

Jones, C. (1996). *Applied software measurement*. New York: McGraw-Hill.

Kay, A. Squeak Etoys Authoring & Media. Retrieved March 2007 from www. squeakland.org

Kieras, D., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391–438.

Klinger, A., & Salingaros, N. A. (2000). A pattern measure. *Environment and Planning: Planning and Design, 27*, 537–547.

Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English.* Providence, RI: Brown University Press.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1–64.

Lewis, C. (in press). Simplicity in cognitive assistive technology: A framework and agenda for research. *International Journal on Universal Access.*

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering* , SE-2 4, 308–320.

McCabe, T. J., & Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the ACM, 32*(12).

Miller, G. A. (1956). The magic number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81–97.

Myers, B. A., McDaniel, R. G., & Kosbie, D. S. Marquise (1993). Creating complete user interfaces by demonstration. *Proceedings of INTERCHI'93: Human Factors in Computing Systems* (pp. 293–300). Amsterdam.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. *Proceedings of INTERCHI'93 Human Factors in Computing Systems* (pp. 206–213). Amsterdam.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic Books.

O'Neill, J., Castellani, S., Grasso, A., Roulland, F., & Tolmie, P. (2005). Representations can be good enough. *ECSCW 2005: Proceedings of Ninth European Conference on Computer-Supported Cooperative Work* (pp. 267–286). Netherlands: Springer.

Ouden, E., den, Lu, Y., Sonnemanns, P., & Brombacher, A. (2006). Quality and reliability problems from a consumer's perspective: An increasing problem overlooked by businesses? *Quality and Reliability Engineering International, 22*(7), 821–838.

Paivio, A., Yuille, J. C., & Madigan, S. A. (1968). Concreteness, imagery and meaningfulness values for 925 words. *Journal of Experimental Psychology Monagraph Supplement, 76*(3)2, 1–25.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer.* New York: Basic Books.

Pelz, D. (2000). *Dave Pelz's putting Bible.* New York: Doubleday.

Phillips, L. H., Bull, R., Adams, E., & Fraser, L. (2002). Positive mood and executive function: Evidence from Stroop and fluency tasks. *Emotion, 2*(1), 12–22.

Pickover, C. (1985). Tusk: A versatile graphics workstation for speech research. *IBM Research Report: RC 11497*. Armonk, NY: IBM.

Rauterberg, M. (1996). A concept to quantify different measures of user interface attributes: A meta-analysis of empirical studies. *Proceedings of the 1996 IEEE International Conference on Systems, Man and Cybernetics* (pp. 2799–2804). Beijing, China.

Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology, 81*, 275–280.

Reisner, P. (1984). Formal grammar as a tool for analyzing ease of use: Some fundamental concepts. In J. Thomas & M. Schneider (Eds.), *Human factors in computer systems* (pp. 53–78). Norwood, NJ: Ablex.

Richards, J. T., & Hanson, V. L. (2004). Web accessibility: A broader view. *Proceedings of the Thirteenth International ACM World Wide Web Conference, WWW2004*, New York: ACM, 2004.

Rubin, D. C. (1980). 51 Properties of 125 words: A unit analysis of verbal behavior. *J. Verbal Learn & Verbal Behavior, 19*, 736–755.

Shmulevich, I., & Povel, D. J. L. (2000). Complexity measures of musical rhythms. In P. W. M. Desain & W. L. Windsor (Eds.), *Rhythm perception and production* (pp. 239–244). Lisse: Swets & Zeitlinger.

Shmulevich, I., Yli-Harja, O., Coyle, E., Povel, D., & Lemstrom, K. (2001). Perceptual issues in music pattern recognition: Complexity of rhythm and key finding. *Computers and Humanities, 35*, 23–35.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society. 106*, 467–482. Philadelphia: American Philosophical Society.

Simon, H. A., & Barenfeld, M. (1969). Information processing analysis of perceptual processes in problem solving. *Psychological Review, 76*, 473–483.

Simon, H. A., & Gilmartin, K. J. (1973). A simulation of memory for chess positions. *Cognitive Psychology, 5*, 29–46.

Smith, M. A., Cottrell, G. W., & Anderson, K. L. (2001). The early word catches the weights. *Advances in Neural Information Processing Systems, 13*, 52–58.

Taylor, R. P., Spehar, B., Wise, J. A., Clifford, C. W. G., Newell, B. R., Hagerhall, C. M., et al. (2005). Perceptual and physiological responses to the visual complexity of Pollock's dripped fractal patterns. *Nonlinear Dynamics, Psychology & Life Sciences, 9*, 89–114.

The Macallan: Speyside distillery—Scotch Whiskey. Retrieved March 5, 2007 from http://www.whiskey-distilleries.info/Macallan_EN.shtml

Thomas, J. C., Danis, C. M., & Lee, A. (2002). Who speaks for wolf? *IBM Research Report: RC22644.* Armonk, NY: IBM.

Thomas, J. C., & Gould, J. D. (1975). A psychological study of Query By Example. *National Computer Conference Proceedings*, New York, *44* , 439–445.

Thomas, J. C. (1976). Quantifiers and question-asking. *IBM Research Report, RC-5886.* Armonk, NY: IBM.

Thomas, J. C. (1983). Psychological issues in the design of database query languages. In M. E. Sime & M. J. Coombs (Eds.), *Designing for human-computer communication* (pp. 173–206). London: Academic Press.

Thomas, J. C., & Ruben, H. (1973, November 8). *Age and mnemonic techniques in paired-associate learning.* Paper presented at the Gerontological Society Meeting, Miami Beach, Florida.

Thomas, J. C., Fozard, J. L., & Waugh, N. C. (1977). Age-related differences in naming latency. *American Journal of Psychology, 90*(30), 499–509.

Underwood, P. (1991). Who speaks for wolf: A Native American learning story. San Anselmo, CA: Tribe of Two Press.

Vintage Cocktails. (2007). Retrieved March 5, 2007, from http://www.vintagecocktails.com/taste.html

Walston, C. E., & Felix, C. P. (1977). A method of programming measurement and estimation. *IBM Systems Journal, 16*, 54–73.

Weinberg, G. M. (1971). *The psychology of computer programming*. New York: Van Nostrand Reinhold.

Werker, J. F., & Tees, R. C. (1984). Cross-language speech perception: evidence of perceptual reorganization during the first year of life. *Infant Behavior and Development, 7*, 49–63.

Zloof, M. (1975). Query by example. *Proceedings of the AFIPS National Computer Conference, 44*, 431–438. Montvale, NJ: AFIPS Press.

·26·

INFORMATION VISUALIZATION

Stuart Card
Xerox PARC

Introduction **510**
Example 1: Finding Videos with the FilmFinder 510
Example 2: Monitoring Stocks with TreeMaps 512
Example 3: Sensemaking with Permutation Matrices 513
What Is Information Visualization? 515
Why Does Visualization Work? 515
Historical Origins 516
The Visualization Reference Model **519**
Mapping Data to Visual Form 519
Data Structures 519
Visual Structures 520
Spatial substrate 520
Marks 521
Connection and enclosure 522
Retinal properties 522
Temporal encoding 522
Expressiveness and Effectiveness 523
Taxonomy of Information Visualizations 523
Simple Visual Structures **524**
1-Variable 525
2-Variables 526
3-Variables and Information Landscapes 526
n-Variables 527
Trees 527
Connection 527
Enclosure 530
Networks 530
Composed Visual Structures **531**
Single-axis composition 531
Double-axis composition 532
Mark composition and case composition 533
Recursive composition 533
Interactive Visual Structures **534**
Dynamic queries 534
Magic lens (movable filter) 534
Overview 1 detail 534
Linking and brushing 534
Extraction and comparison 536
Attribute explorer 536
Focus 1 Context Attention-Reactive Abstractions **536**
Data-Based Methods 536
Filtering 536
Selective aggregation 537
View-Based Methods 537
Micro-macro readings 537
Highlighting 537
Visual transfer functions 537
Perspective distortion 538
Alternate geometries 539
Sensemaking With Visualization **539**
Knowledge Crystallization 539
Acquire information 540
Make sense of it 540
Create something new 540
Act on it 540
Levels for Applying Information Visualization 540
Acknowledgment **542**
References **542**

INTRODUCTION

The working mind is greatly leveraged by interaction with the world outside it. A conversation to share information, a grocery list to aid memory, a pocket calculator to compute square roots—all effectively augment a cognitive ability otherwise severely constrained by what is in its limited knowledge, by limited attention, and by limitations on reasoning. But the most profound leverage on cognitive ability is the ability to invent new representations, procedures, or devices that augment cognition far beyond its unaided biological endowment—and bootstrap these into even more potent inventions.

This chapter is about one class of inventions for augmenting cognition, collectively called "information visualization." Other senses could be employed in this pursuit—audition, for example, or a multi-modal combination of senses—the broader topic is really *information perceptualization*; however, in this chapter, we restrict ourselves to visualization. Visualization employs the sense with the most information capacity; recent advances in graphically agile computers have opened opportunities to exploit this capacity, and many visualization techniques have now been developed. A few examples suggest the possibilities.

Example 1: Finding Videos with the FilmFinder

The use of information visualization for finding things is illustrated by the FilmFinder (Ahlberg & Shneiderman, 1994a, 1994b). Unlike typical movie-finder systems, the FilmFinder is organized not around searching with keywords, but rather around rapid browsing and reacting to collections of films in the database Figure 26.1 shows a scattergraph of 2000 movies, plotting rated quality of the movie as a function of year when it was released. Color differentiates type of movies—comedy from drama and the like. The display provides an overview, the entire universe of all the movies, and some general features of the collection. It is visually apparent, for example, that a good share of the movies in the collection were released after 1965, but also that there are movies going back as far as the 1920s. Now the viewer "drills down" into the collection by using the sliders in the interface to show only movies with Sean Connery that are between 1 and 4½ hours in length (Fig. 26.2). As the sliders are moved, the display zooms in to show about 20 movies. It can be seen that these movies were made between 1960 and 1995, and all have a quality rating higher than 4. Since there is now room on the display, titles of the movies appear. Experimentation with the slider shows that restricting maximum length to 2 hours cuts out few interesting movies. The viewer chooses the highly rated movie, "Murder on the Orient Express" by double-clicking on its marker. Up pop details in a box (Fig. 26.3) giving names of other actors in the movie and more information. The viewer is interested in whether two of these actors, Anthony Perkins and Ingrid Bergman, have appeared together in any other movies. The viewer selects their names in the box, and then requests another search (Fig. 26.4). The result is a new display of two movies. In addition to the movie the viewer knew about, there is one other movie, a drama entitled "Goodbye,

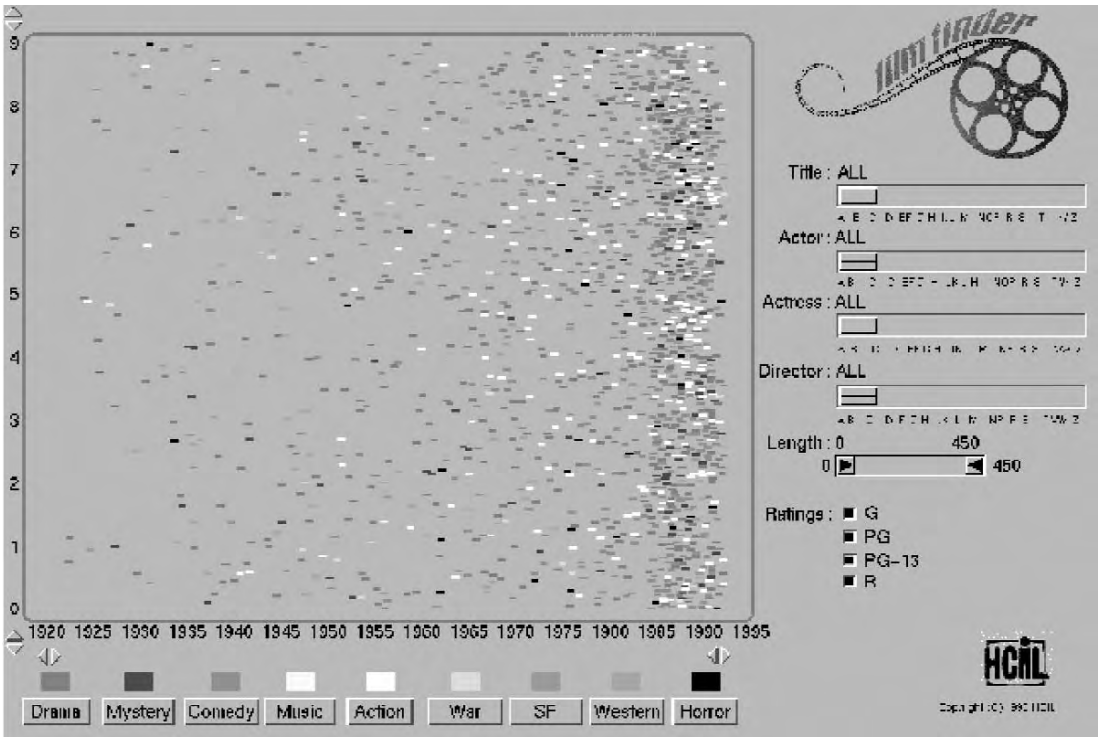


FIGURE 26.1. FilmFinder overview scattergraph. Courtesy University of Maryland.

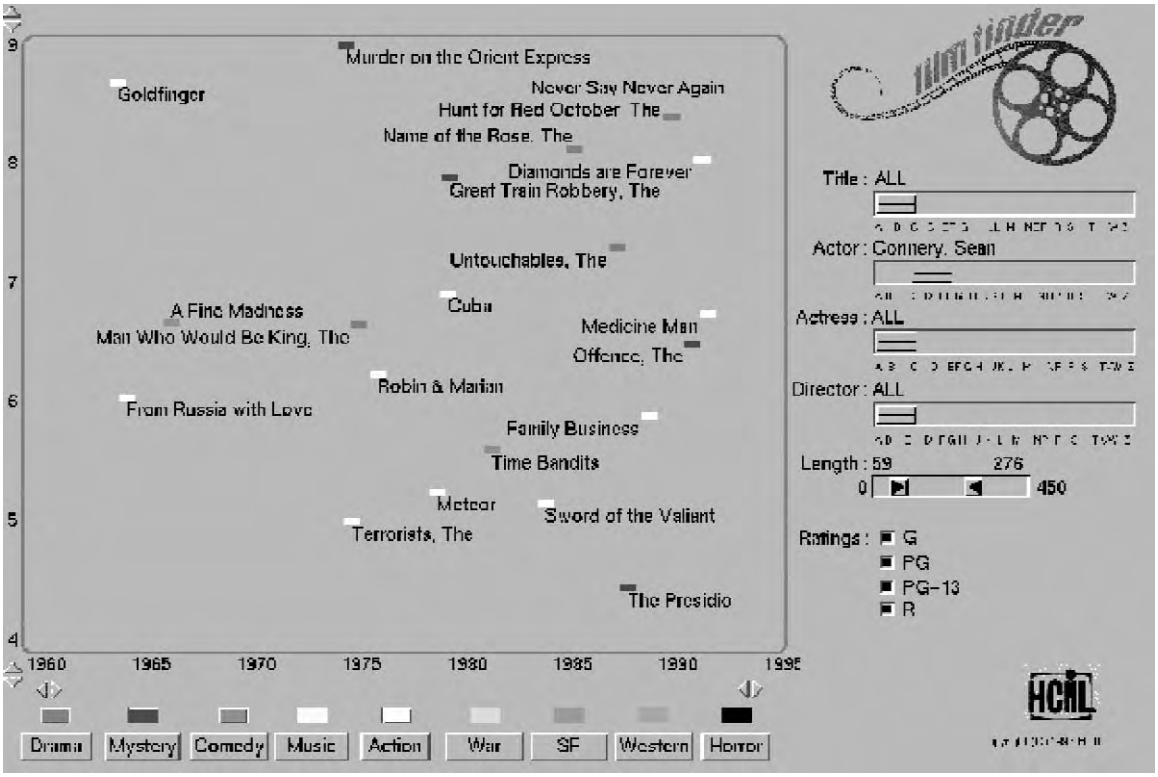


FIGURE 26.2. FilmFinder scattergraph zoom-in. Courtesy University of Maryland.

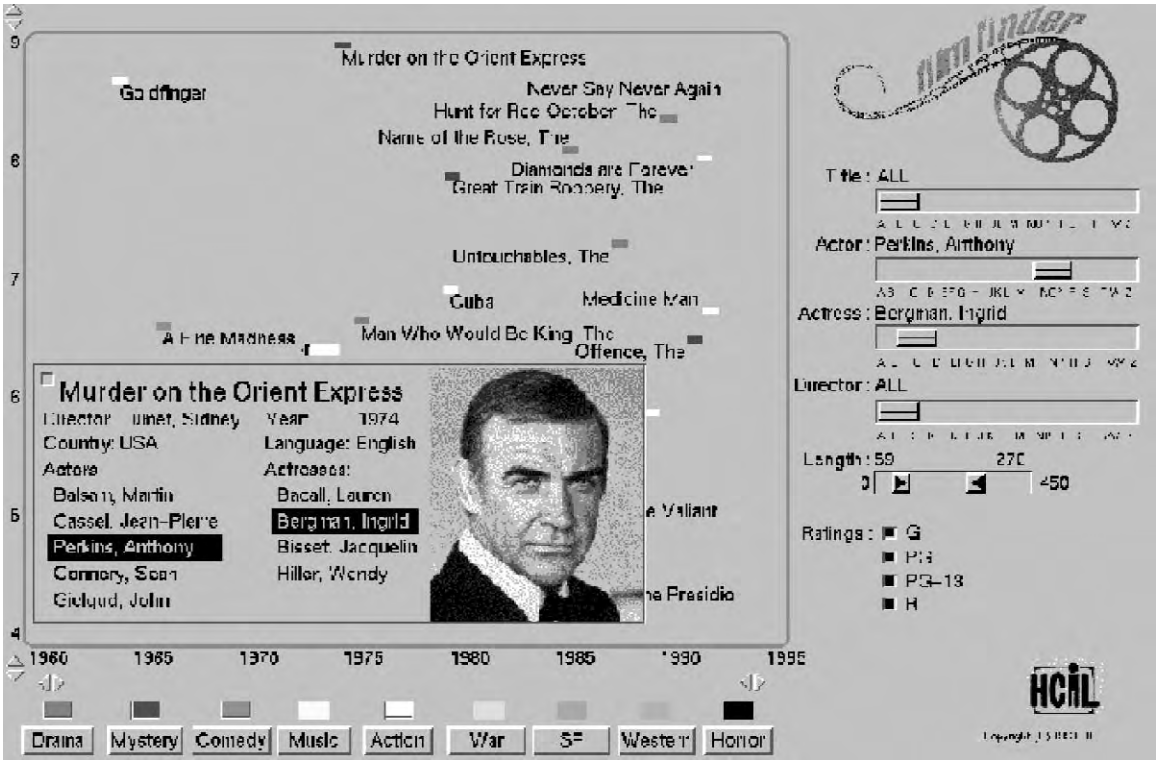


FIGURE 26.3. FilmFinder details on demand. Courtesy University of Maryland.

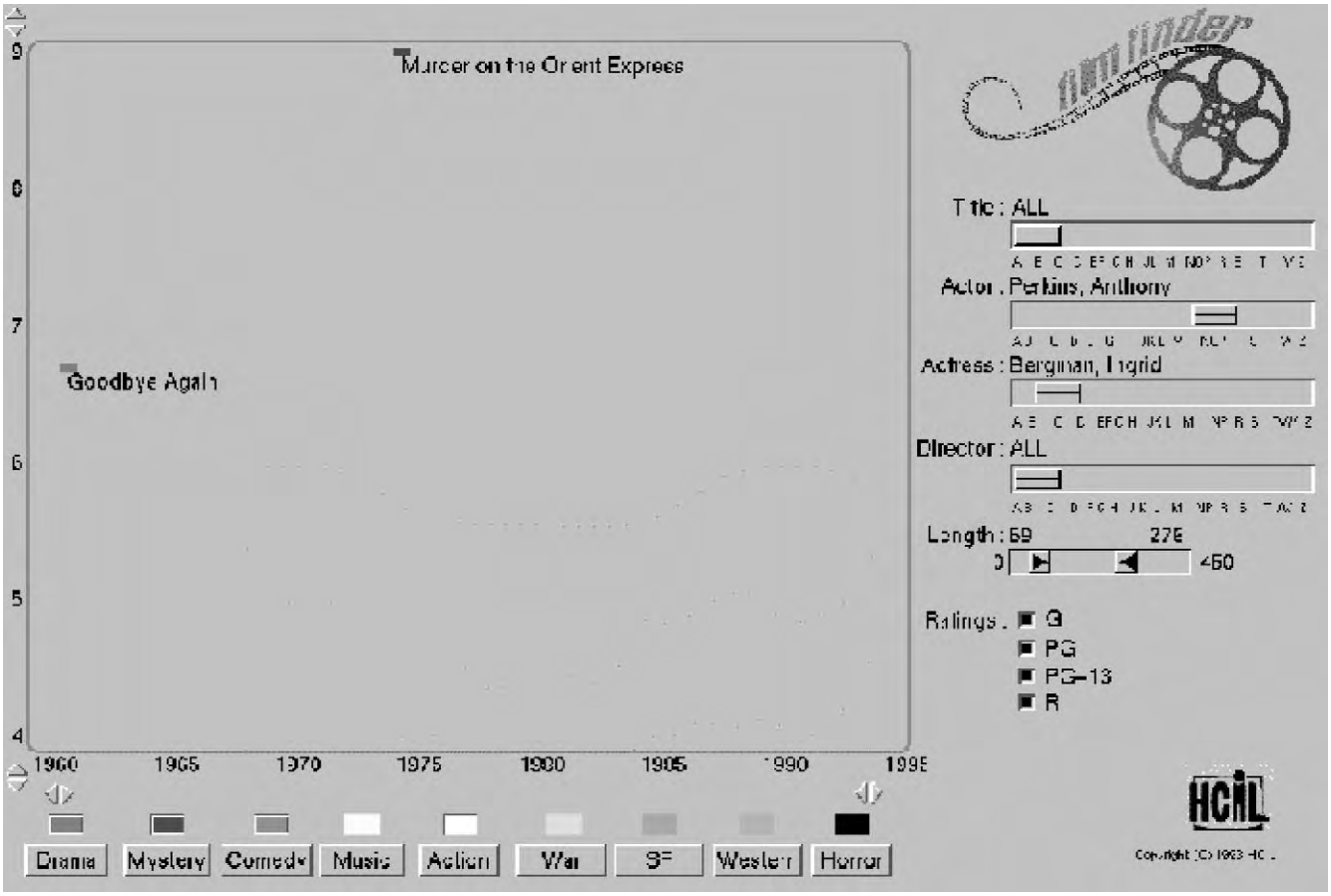


FIGURE 26.4. FilmFinder retrieval by example. Courtesy University of Maryland.

Again," made around 1960. The viewer is curious about this movie and decides to watch it.

Information visualization has allowed a movie viewer in a matter of seconds to find a movie he or she could not have specified at the outset. To do this, the FilmFinder employed several techniques from information visualization: (a) an *overview* of the collection showing its structure; (b) *dynamic queries*, in which the visualization seems to change instantaneously with control manipulations; (c) *zooming in* by adding restrictions to the set of interest; (d) *details on demand*, in which the user can display temporarily detail about an individual object, and (e) *retrieval by example*, in which selected attributes of an individual object are used to specify a new retrieval set.

Example 2: Monitoring Stocks with TreeMaps

Another example of information visualization is the TreeMap visualization on the SmartMoney.com website,[1] which is shown in Fig. 26.5(a). Using this visualization, an investor can monitor more than 500 stocks at once, with data updated every 15 minutes. Each colored rectangle in the figure is a company. The size of the rectangle is proportional to its market capitalization. Color of the rectangle shows movement in the stock price. Bright yellow corresponds to about a 6% increase in price, bright blue to about a 6% decrease in price. Each business sector is identified with a label like "Communications." Those items marked with a letter *N* have an associated news item.

In this example, the investor's task is to monitor the day's market and notice interesting developments. In Fig. 26.5(a), the investor has moved the mouse over one of the bright yellow rectangles, and a box identifying it as Erickson, with a +9.28% gain for the day, has popped up together with other information. Clicking on a box gives the investor a popup menu for selecting even more detail. The investor can either click to go to World Wide Web links on news or financials, or drill down, for example, to the sector (Fig. 26.5[b]), or down further to individual companies in the software part of the technology sector (Fig. 26.5[c]). The investor is now able to immediately note interesting relationships. The software industry is now larger than

[1]www.smartmoney.com

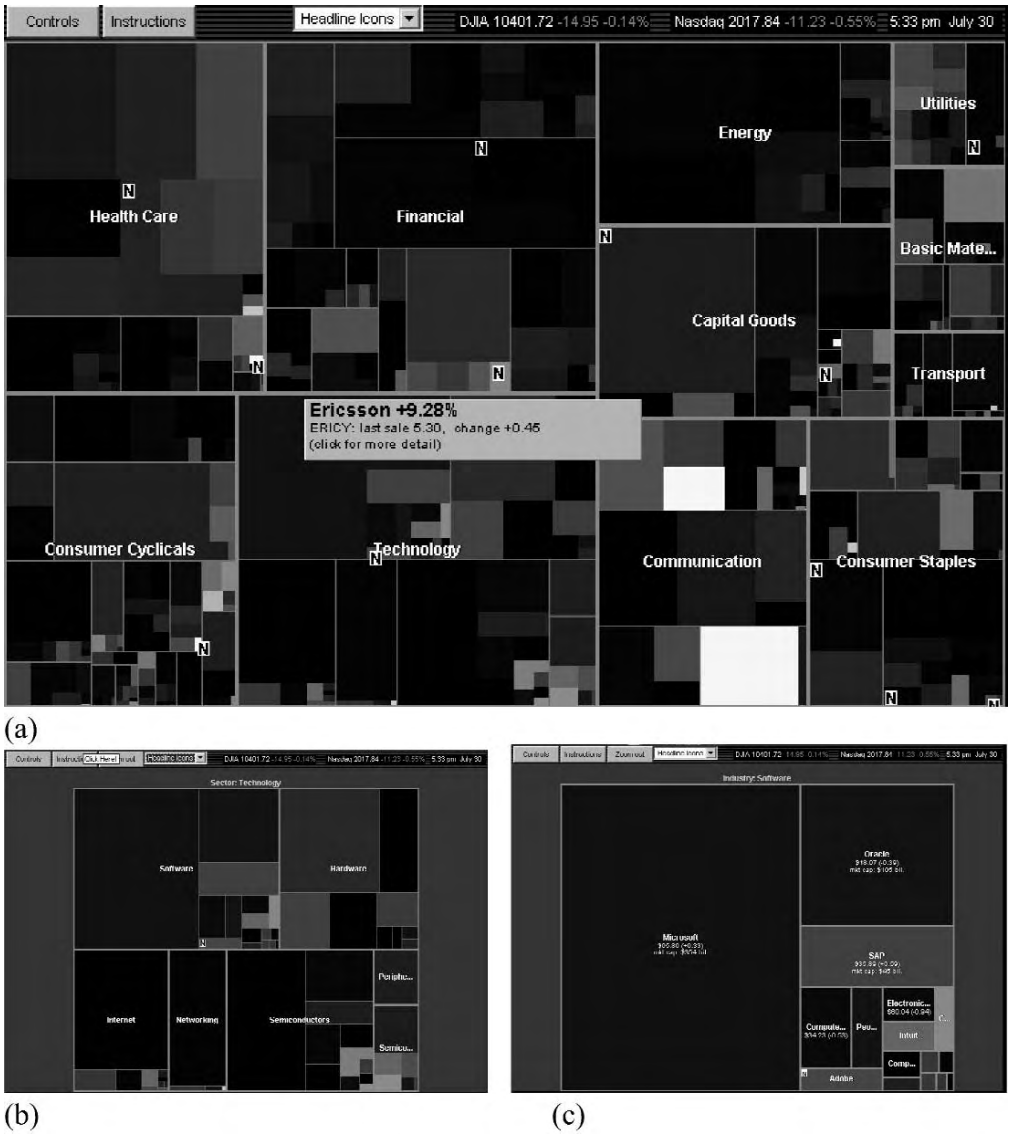


FIGURE 26.5. TreeMap of daily stock prices. Courtesy SmartMoney.com.

the hardware industry, for example, and despite a recent battering at the time of this figure, the Internet industry is also relatively large. Microsoft is larger than all the other companies in its industry combined. Selecting a menu item to look at year-to-date gains (Fig. 26.6), the investor immediately notes interesting patterns: Microsoft stock shows substantial gains, whereas Oracle is down; Dell is up, but Compaq is down; Tiny Advanced Micro is up, whereas giant Intel is neutral. Having noticed these relationships, the investor drills down to put up charts or analysts positions for companies whose gains in themselves, *or in relation to a competitor*, are interesting. For example, the investor is preparing a report on the computer industry for colleagues and notices how AMD is making gains against Intel, or how competition for the Internet is turning into a battle between Microsoft and AOL/Time Warner.

Example 3: Sensemaking with Permutation Matrices

As a final information visualization example, consider the case proposed by Bertin (1977/1981) of a hotel manager who wants to analyze hotel occupancy data (Table 26.1) to increase her return. In order to search for meaningful patterns in her data, she represents it as a permutation matrix (Fig. 26.7[a]. A permutation matrix is a graphic rendition of a cases x variables display. In Fig. 26.7(a), each cell of Table 26.1 is a small bar of a bar chart. The bars for cells below the mean are white; those above the bar are black. By permuting rows and columns, patterns emerge that lead to making sense of the data.

In Fig. 26.7(a), the set of months, which form the cases, are repeated to reveal periodic patterns across the end of the cycle. By visually comparing the pairs of rows, one can find rows

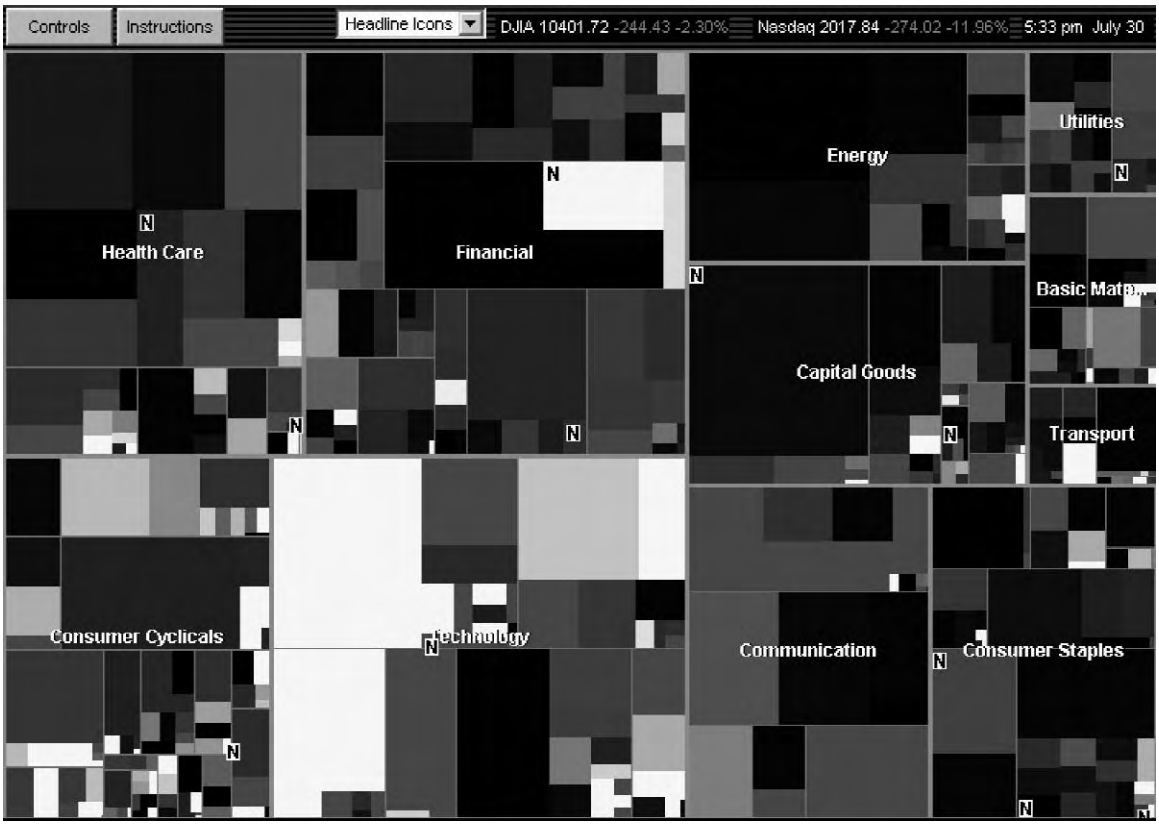


FIGURE 26.6. TreeMap of year-to-date stock prices. Courtesy SmartMoney.com.

TABLE 26.1. Data for Hotel Occupancy (Based on Bertin (1977/1981))

ID	VARIABLE	JAN	FEB	MAR	APR	MAY	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
1	% Female	26	21	26	28	20	20	20	20	20	40	15	40
2	% Local	69	70	77	71	37	36	39	39	55	60	68	72
3	% USA	7	6	3	6	23	14	19	14	9	6	8	8
4	% South America	0	0	0	0	8	6	6	4	2	12	0	0
5	% Europe	20	15	14	15	23	27	22	30	27	19	19	17
6	% M.East/Africa	1	0	0	8	6	4	6	4	2	1	0	1
7	% Asia	3	10	6	0	3	13	8	9	5	2	5	2
8	% Businessmen	78	80	85	86	85	87	70	76	87	85	87	80
9	% Tourists	22	20	15	14	15	13	30	24	13	15	13	20
10	% Direct Reservations	70	70	75	74	69	68	74	75	68	68	64	75
11	% Agency Reservations	20	18	19	17	27	27	19	19	26	27	21	15
12	% Air Crews	10	12	6	9	4	5	7	6	6	5	15	10
13	% Under 20	2	2	4	2	2	1	1	2	2	4	2	5
14	% 20–35	25	27	37	35	25	25	27	28	24	30	24	30
15	% 35–55	48	49	42	48	54	55	53	51	55	46	55	43
16	% Over 55	25	22	17	15	19	19	19	19	19	20	19	22
17	Price of rooms	163	167	166	174	152	155	145	170	157	174	165	156
18	Length of stay	1.7	1.7	1.7	1.91	1.9	2	1.54	1.6	1.73	1.82	1.66	1.44
19	% Occupancy	67	82	70	83	74	77	56	62	90	92	78	55
20	Conventions	0	0	0	1	1	1	0	0	1	1	1	1

that are similar. These are reordered and grouped (Fig. 26.7[b]). By this means, it is discovered that there seem to be two patterns of yearly variation. One pattern in Fig. 26.7(b) is semiannual, dividing the year into the cold months of October through April and the warm months of May through September. The other pattern breaks the year into four distinct regions. We have thus found the beginnings of a *schema*—that is, a framework in terms of which we can encode the raw data and describe it in a more compact language. Instead of talking about the events of the year in terms of individual months, we can now talk in terms of two series of periods, the semiannual one, and the four distinct periods. As we do so, there is a *residue* of information not included as part of our descriptive language. Sensemaking proceeds by the *omission and recoding of information into more compact form* (see Resnikoff, 1989). This residue of information may be reduced by finding a better or more articulated schema, or it may be left as noise. Beyond finding the basic patterns in the data, the hotel manager wants to make sense of the data relative to a purpose: she wants to increase the occupancy of the hotel. Therefore, she has also permuted general indicators of activity in Fig. 26.7(b), such as % Occupancy and Length of Stay, to the top of the diagram and put the rows that correlate with these below them. This reveals that Conventions, Businessmen, and Agency Reservations, all of which generally have to do with convention business, are associated with higher occupancy. This insight comes from the match in patterns *internal* to the visualization; it also comes from noting why these variables might correlate as a consequence of factors *external* to the visualization. She also discovers that marked differences exist between the winter and summer guests during the slow periods. In winter, there are more local guests, women, and age differences. In summer, there are more foreign tourists and less variation in age.

This visualization was useful for sensemaking on hotel occupancy data, but it is too complicated to communicate the high points. The hotel manager therefore creates a simplified diagram, Fig. 26.7(c). By graying some of the bars, the main points are more readily graspable, while still preserving the data relations. A December convention, for example, does not seem to have the effect of the other conventions in bringing in guests. It is shown in gray as residue in the pattern. The hotel manager suggests moving the convention to another month, where it might have more effect on increasing the occupancy of the hotel.

What Is Information Visualization?

The FilmFinder, the TreeMap, and the permutation matrix hotel analysis are all examples of the use of information visualization. We can define information visualization as "the use of computer-supported, interactive, visual representations of abstract data in order to amplify cognition" (Card, Mackinlay, & Shneiderman, 1999).

Information visualization needs to be distinguished from related areas: *scientific visualization* is like information visualization, but it is applied to scientific data and typically is physically based. The starting point of a natural geometrical substrate for the data, whether the human body or earth geography, tends to emphasize finding a way to make visible the invisible (say, velocity of air flow) within an existing spatial framework. The chief problem for information visualization, in contrast, is often finding an effective mapping between abstract entities and a spatial representation. Both information visualization and scientific visualization belong to the broader field of *data graphics*, which is the use of abstract, nonrepresentational visual representations to amplify cognition. Data graphics, in turn, is part of *information design*, which concerns itself with external representations for amplifying cognition. At the highest level, we could consider information design a part of *external cognition*, the uses of the external world to accomplish some cognitive process. Characterizing the purpose of information visualization as *amplifying cognition* is purposely broad. Cognition can be the process of writing a scientific paper or shopping on the Internet for a cell phone. Generally, it refers to the intellectual processes in which information is obtained, transformed, stored, retrieved, and used. All of these can be advanced generally by means of external cognition, and in particular by means of information visualization.

Why Does Visualization Work?

Visualization aids cognition not because of some mystical superiority of pictures over other forms of thought and communication, but rather because visualization helps the user by making the world outside the mind a resource for thought in specific ways. We list six groups of these in Table 26.2 (Card et al., 1999): Visualization amplifies cognition by (a) increasing the memory and processing resources available to the users, (b) reducing search for information, (c) using visual representations to enhance the detection of patterns, (d) enabling perceptual inference operations, (e) using perceptual attention mechanisms for monitoring, and (f) by encoding information in a manipulable medium. The FilmFinder, for example, allows the representation of a large amount of data in a small space in a way that allows patterns to be perceived visually in the data. Most important, the method of instantly responding in the display to the dynamic movement of the sliders allowed users to rapidly explore the multidimensional space of films. The TreeMap of the stock market allows monitoring and exploration of many equities. Again, much data is represented in little space. In this case, the display manages the user's attention, drawing it to those equities with unusually large changes, and supplying the means to drill down into the data to understand why these movements may be happening. In the hotel management case, the visual representation makes it easier to notice similarities of behavior in a multidimensional attribute space, then to cluster and rerepresent these. The final product is a compact (and simplified) representation of the original data that supports a set of forward decisions. In all of these cases, visualization allows the user to (a) examine a large amount of information, (b) keep an overview of the whole while pursuing details, (c) keep track of (by using the display as an external working memory) many things, and (d) produce an abstract representation of a situation through the omission and recoding of information.

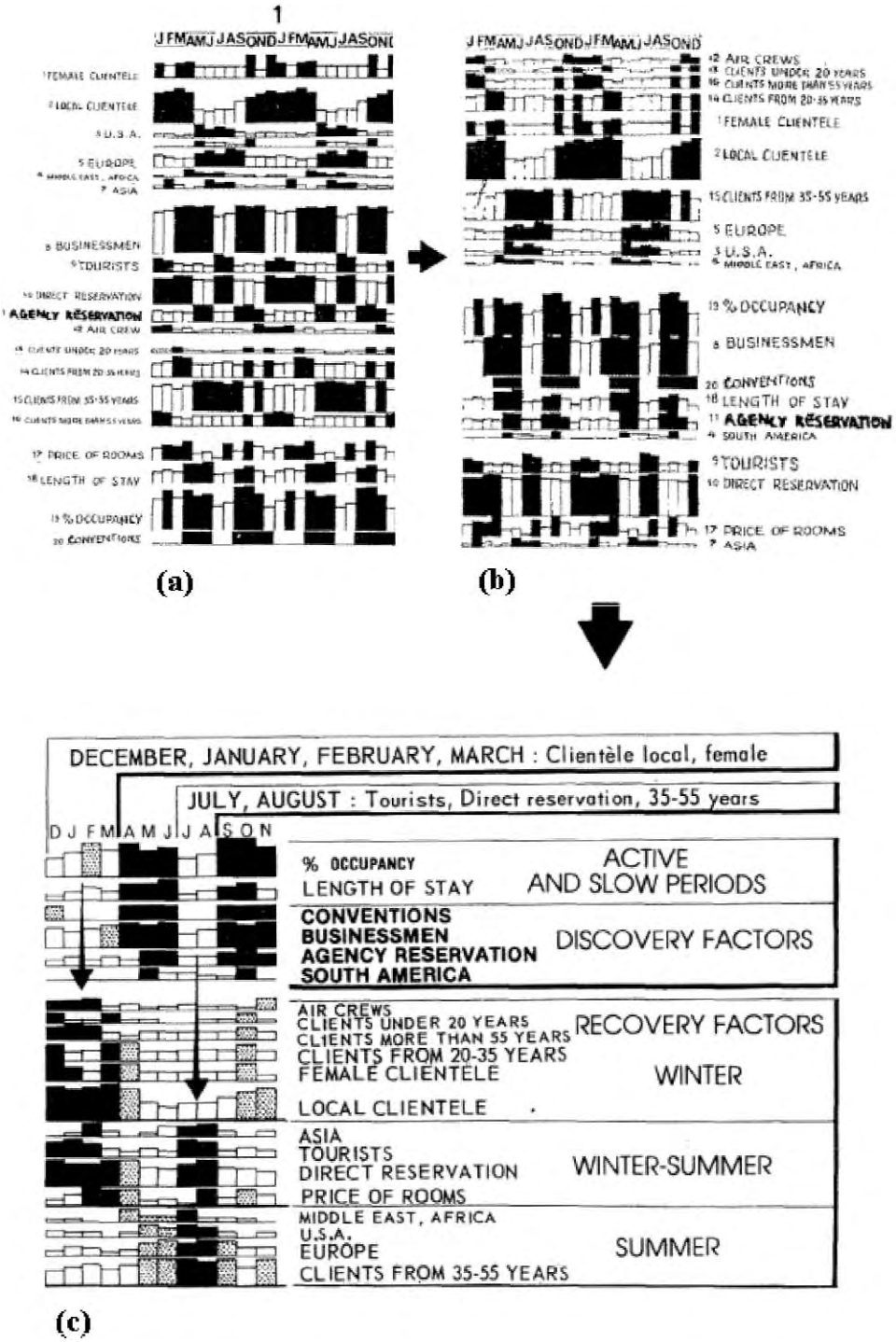


FIGURE 26.7. Permutation matrix representation of hotel data from (Berlin, 1977/1981). (a) Initial matrix of variables. (b) Permuted matrix to group like patterns together.

Historical Origins

Drawn visual representations have a long history. Maps go back millennia. Diagrams were an important part of Euclid's books on geometry. Science, from earliest times, used diagrams to (a) record observations, (b) induct relationships, (c) explicate methodology of experiments, and (d) classify and conceptualize phenomena (for a discussion, see Robin, 1992). For example, Fig. 26.8 is a hand-drawn illustration, in Newton's first scientific publication, illustrating how white light is really composed of

TABLE 26.2. How Information Visualization Amplifies Cognition

1. Increased Resources	
High-bandwidth hierarchical interaction	Human moving gaze system partitions limited channel capacity so that it combines high spatial resolution and wide aperture in sensing the visual environments (Larkin & Simon, 1987).
Parallel perceptual processing	Some attributes of visualizations can be processed in parallel compared to text, which is serial.
Offload work from cognitive to perceptual system	Some cognitive inferences done symbolically can be recoded into inferences done with simple perceptual operations (Larkin & Simon, 1987).
Expanded working memory	Visualizations can expand the working memory available for solving a problem (Norman, 1993).
Expanded storage of information	Visualizations can be used to store massive amounts of information in a quickly accessible form (e.g., maps).
2. Reduced Search	
Locality of processing.	Visualizations group information used together reducing search (Larkin & Simon, 1987).
High data density.	Visualizations can often represent a large amount of data in a small space (Tufte, 1983).
Spatially-indexed addressing.	By grouping data about an object, visualizations can avoid symbolic labels (Larkin & Simon, 1987).
3. Enhanced Recognition of Patterns	
Recognition instead of recall.	Recognizing information generated by a visualization is easier than recalling that information by the user.
Abstraction and aggregation	Visualizations simplify and organize information, supplying higher centers with aggregated forms of information through abstraction and selective omission (Card, Robertson, & Mackinlay, 1991); (Resnikoff, 1989).
Visual schemata for organization	Visually organizing data by structural relationships (e.g., by time) enhances patterns.
Value, relationship, trend	Visualizations can be constructed to enhance patterns at all three levels (Bertin, 1967/1983).
4. Perceptual Inference	
Visual representations make some problems obvious	Visualizations can support a large number of perceptual inferences that are very easy for humans (Larkin & Simon, 1987).
Graphical computations	Visualizations can enable complex specialized graphical computations (Hutchins, 1996).
5. Perceptual Monitoring	Visualizations can allow for the monitoring of a large number of potential events if the display is organized so that these stand out by appearance or motion.
6. Manipulable medium	Unlike static diagrams, visualizations can allow exploration of a space of parameter values and can amplify user operations.

Source: Card, Mackinlay, & Shneiderman, 1999.

many colors. Sunlight enters from the window at right and is refracted into many colors by a prism. One of these colors can be selected (by an aperture in a screen) and further refracted by another prism, but the light stays the same color, showing that it has already been reduced to its elementary components. As in Newton's illustration, early scientific and mathematical diagrams generally had a spatial, physical basis and were used to reveal the hidden, underlying order in that world.

Surprisingly, diagrams of abstract, nonphysical information are apparently rather recent. Tufte (1983) dates abstract diagrams to (Playfair, 1786) in the 18th century. Figure 26.9 is one of Playfair's earliest diagrams. The purpose was to convince readers that English imports were catching up with imports. Starting with Playfair, the classical methods of plotting data were developed—graphs, bar charts, and the rest.

Recent advances in the visual representation of abstract information derive from several strands that became intertwined. In 1967, Bertin (1967/1983, 1977/1981), a French cartographer published his theory of *The Semiology of Graphics.* This theory identified the basic elements of diagrams and their combination. Tufte (1983, 1990, 1997), from the fields of visual design and data graphics, published a series of seminal books that set forth principles for the design of data graphics and emphasized maximizing the density of useful information. Both Bertin's and Tufte's theories became well known and influential. Meanwhile, within statistics, Tukey (1977) began a movement on exploratory data analysis. His emphasis was not on the quality of graphical presentation, but on the use of pictures to give rapid, statistical insight into data relations. For example, "box and whisker plots" allowed an analyst to get a rapid characterization of data distributions. Cleveland and McGill (1988) wrote an influential book, *Dynamic Graphics for Statistics*, explicating new visualizations of data with particular emphasis on the visualization of multidimensional data.

In 1985, NSF launched an initiative on *scientific visualization* (McCormick & DeFanti, 1987). The purpose of this initiative was to use advances in computer graphics to create a new class of analytical instruments for scientific analysis, especially as a tool for comprehending large, newly produced datasets in the geophysical and biological sciences. Meanwhile, the computer graphics and artificial intelligence communities were interested in the automatic design of visual presentations of data. Mackinlay's (1986a, 1986b) thesis APT formalized Bertin's design theory, added psychophysical data, and used these to build a system for automatically generating diagrams of data, tailored for some purpose. Roth and Mattis (1990) built a system to do more complex visualizations, such as some of those from Tufte. Casner (1991) added a representation of tasks. This community was interested not so much in the quality of the graphics as in the automation of the match between data characteristics, presentational purpose, and graphical presentation. Finally, the user interface community saw advances in graphics hardware opening the possibility of a new generation of user interfaces. The first use of the term "information visualization" was probably in

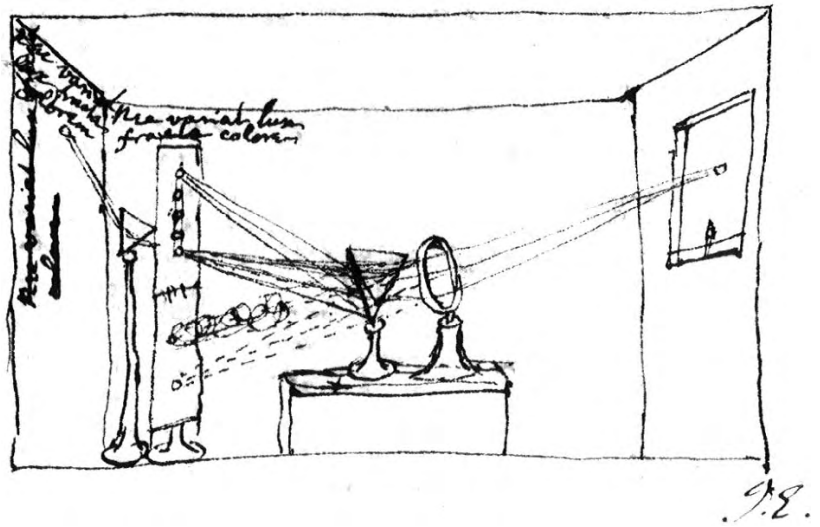


FIGURE 26.8. Newton's optics illustration (from Robin, 1992).

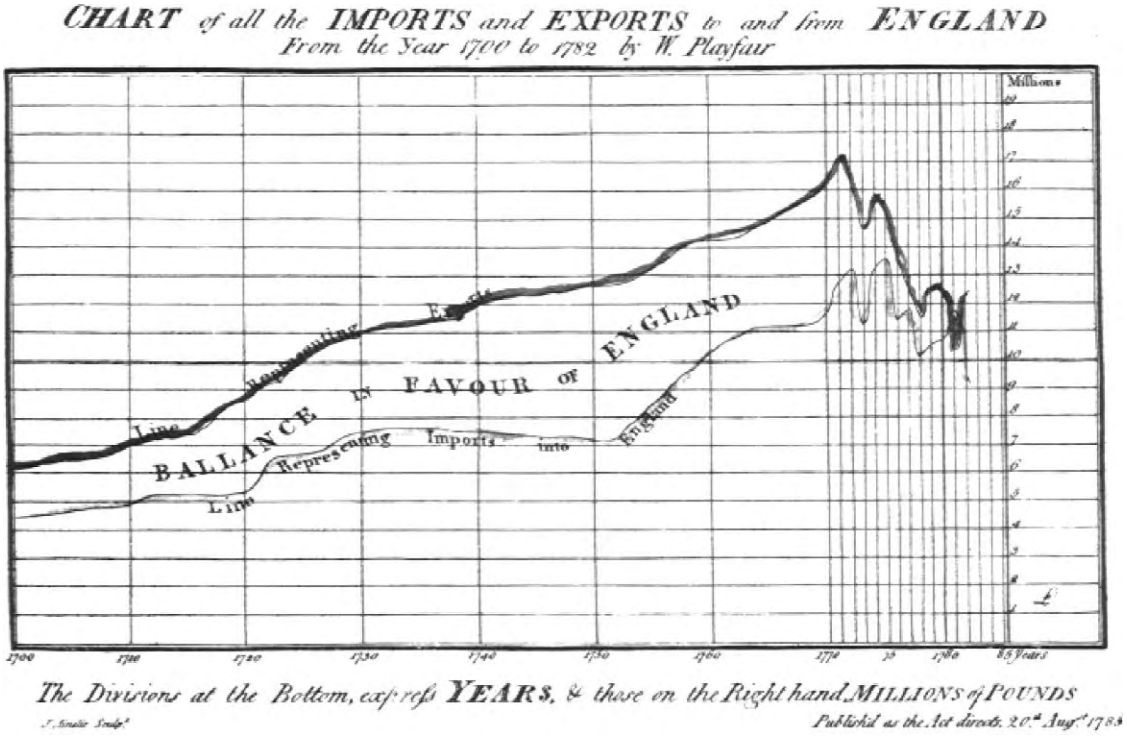


FIGURE 26.9. Playfair's charts of English imports and exports, from (Tufte, 1983).

Robertson, Card, and Mackinlay (1989). Early studies in this community focused on user interaction with large amounts of information: Feiner and Beshers (1990) presented a method, worlds within worlds, for showing six-dimensional financial data in an immersive virtual reality. Shneiderman (1992) developed a technique called "dynamic queries" for interactively selecting subsets of data items and TreeMaps, a space-filling representation for trees. Robertson, Card, and Mackinlay (1993) presented ways of using animation and distortion to interact with large data sets in a system called the Information Visualizer, which used *focus + context* displays to nonuniformly present large amounts of information. The emphasis for these studies was on the means for cognitive amplification, rather than on the quality of the graphics presentations.

The remainder of this chapter will concentrate on the techniques that have been developed for mapping abstract information to interactive visual form to aid some intellectual task. The perceptual foundations of this effort are beyond the scope of this chapter, but are covered in Ware (2000). Further details on information visualization techniques are addressed in a text by Spence (2000). The classic papers in information visualization are collected in Card et al. (1999).

THE VISUALIZATION REFERENCE MODEL

Mapping Data to Visual Form

Despite their seeming variability, information visualizations can be systematically analyzed. Visualizations can be thought of as adjustable mappings from data to visual form to the human perceiver. In fact, we can draw a simple Visualization Reference Model of these mappings (Fig. 26.10). Arrows follow from *Raw Data* (data in some idiosyncratic format) on the left, though a set of *Data Transformations* into *Data Tables* (canonical descriptions of data in a variables x cases format extended to include metadata). The most important mapping is the arrow from Data Tables to *Visual Structures* (structures that combine values an available vocabulary of visual elements—spatial substrates, marks, and graphical properties). Visual Structures can be further transformed by *View Transformations*, such as visual distortion or 3D viewing angle, until it finally forms a *View* that can be perceived by human users. Thus, Raw Data might start out as text represented as indexed strings or arrays. These might be transformed into *document vectors*, normalized vectors in a space with dimensionality as large as the number of words. Document vectors, in turn, might be reduced by multidimensional scaling to create the analytic abstraction to be visualized, expressed as a Data Table of x, y, z coordinates that could be displayed. These coordinates might be transformed into a Visual Structure—that is, a surface on an information landscape—which is then viewed at a certain angle.

Similar final effects can be achieved by transformations at different places in the model: When a point is deleted from the visualization, has the point been deleted from the dataset? Or is it still in the data merely not displayed? Chi and Riedl (1998) called this the *view-value distinction*, and it is an example of just one issue where identifying the locus of a transformation using the Visualization Reference Model helps to avoid confusion.

Information visualization is about the not just creation of visual images, but also the interaction with those images in the service of some problem. In the Visualization Reference Model, another set of arrows flow back from the human at the right into the transformations themselves, indicating the adjustment of these transformations by user-operated controls. It is the rapid reciprocal reaction between the generation of images by machine and the selection and parametric adjustment of those images, giving rise to new images that gives rise to the attractive power of interactive information visualization.

Data Structures

It is convenient to express Data Tables as tables of objects and their attributes, as in Table 26.3. For example, in the FilmFinder, the basic objects (or "cases") are films. Each film is associated with a number of attributes or variables, such as title, stars, year of release, genre type, and so forth. The vertical double black line in the table separates data in the table to the left of the line

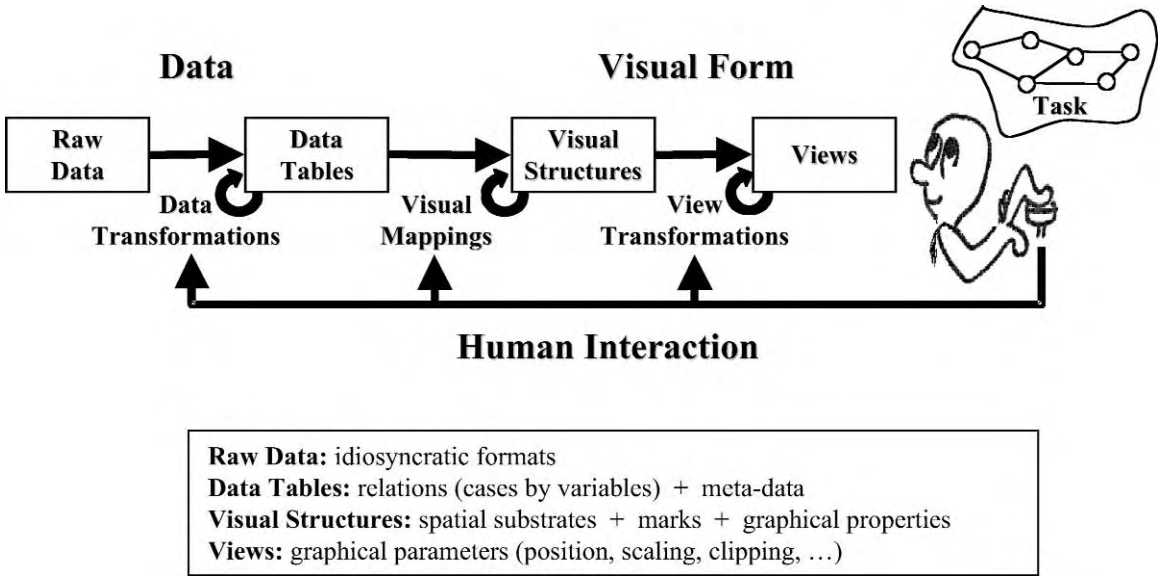


FIGURE 26.10. Reference model for visualization (Card et al., 1999). Visualization can be described as the mapping of data to visual form that supports human interaction in a workplace for visual sense making.

TABLE 26.3. A Data Table About Films

FilmID	230	105	540	. . .
Title	Goldfinger	Ben Hur	Ben Hur	. . .
Director	Hamilton	Wyler	Niblo	. . .
Actor	Connery	Heston	Novarro	. . .
Actress	Blackman	Harareet	McAvoy	. . .
Year	1964	1959	1926	. . .
Length	112	212	133	. . .
Popularity	7.7	8.2	7.4	
Rating	PG	G	G	. . .
FilmType	Action	Action	Drama	. . .

Source: (Card et al., 1999).

from the metadata, expressed as variable names, to the left of the line. The horizontal black line across the table separates input variables from output variables—that is, the table can be thought of as a function.

$$f(\text{input variables}) = \text{output variables}.$$

So,

$$\textit{Year}\ (\textit{FilmID} = 105) = 1959.$$

Variables imply a scale of measurement, and it is important to keep these straight. The most important to distinguish are

N = *Nominal (are only = or ≠ to other values)*
O = *Ordinal* (obeys a < relation)
Q = *Quantitative* (can do arithmetic on them)

A nominal variable N is an unordered set, such as film titles {Goldfinger, Ben Hur, Star Wars}. An ordinal variable O is a tuple (ordered set), such as film ratings ⟨G, PG, PG-13, R⟩. A quantitative variable Q is a numeric range, such as film length [0, 360].

In addition to the three basic types of variables, subtypes represent important properties of the world associated with specialized visual conventions. We sometimes distinguish the subtype *Quantitative Spatial* (Q_s) for intrinsically spatial variables common in scientific visualization and the subtype *Quantitative Geographical* (Q_g) for spatial variables that are specifically geophysical coordinates. Other important subtypes are similarity metrics *Quantitative Similarity* (Q_m), and the temporal variables *Quantitative Time* (Q_t) and *Ordinal Time* (O_t). We can also distinguish Interval Scales (I) (like Quantitative Scales, but since there is not a natural zero point, it is not meaningful to take ratios). An example would be dates. It is meaningful to subtract two dates (June 5, 2002 – June 3, 2002 = 2 days), but it does not make sense to divide them (June 5, 2002 ÷ June 23, 2002 = Undefined). Finally, we can define an *Unstructured* Scale (U), whose only value is present or absent (e.g., an error flag). The scales are summarized in Table 26.4.

Scale types can be altered by transformations, and this practice is sometimes convenient. For example, quantitative variables can be mapped by data transformations into ordinal variables

$$Q \rightarrow O$$

by dividing them into ranges. For example, film lengths [0, 360] minutes (type Q) can be broken into the ranges (type O),

$$[0, 360]\ \text{minutes} \rightarrow \langle \textsc{Short}, \textsc{Medium}, \textsc{Long} \rangle.$$

This common transformation is called "classing," because it maps values onto classes of values. It creates an accessible summary of the data, although it loses information. In the other direction, nominal variables can be transformed to ordinal values

$$N \rightarrow O$$

based on their name. For example, film titles {Goldfinger, Ben Hur, Star Wars} can be sorted lexicographically

$$\{\textsc{Goldfinger}, \textsc{Ben Hur}, \textsc{Star Wars}\} \rightarrow \langle \textsc{Ben Hur}, \textsc{Goldfinger}, \textsc{Star Wars} \rangle.$$

Strictly speaking, we have not transformed their values, but in many uses (e.g., building alphabetically arranged dictionaries of words or sliders in the FilmFinder), we can act as if we had.

Variable scale types form an important class of metadata that, as we shall see, is important for proper information visualization. We can add scale type to our Data Table in Table 26.3 together with cardinality or range of the data to give us essentially a codebook of variables as in Table 26.5.

Visual Structures

Information visualization maps data relations into visual form. At first, it might seem that a hopelessly open set of visual forms can result. Careful reflection, however, reveals what every artist knows: that visual form is subject to strong constraints. Visual form that reflects the systematic mapping of data relations onto visual form, as in information visualization or data graphics, is subject to even more constraints. It is a genuinely surprising fact, therefore, that most information visualization involves the mapping data relations onto only a half dozen components of visual encoding:

1. *Spatial substrate*
2. *Marks*
3. *Connection*
4. *Enclosure*
5. *Retinal properties, or*
6. *Temporal encoding*

Of these mappings, the most powerful is how data are mapped onto the spatial substrate—that is, how data are mapped into spatial position. In fact, one might say that the design of an information visualization consists first of deciding which variables are going to get the spatial mappings, and then how the rest of the variables are going to make do with the coding mappings that are left.

Spatial substrate. As we have just said, the most important choice in designing an information visualization is which variables are going to map onto spatial position. This decision

TABLE 26.4. Classes of Data and Visual Elements

	Data Classes		*Visual Classes*	
Class	Description	Example	Description	Example
U	U*nstructured* (can only distinguish presence or absence)	ErrorFlag	U*nstructured* (no axis, indicated merely whether something is present or absent)	Dot
N	N*ominal* (can only distinguish whether two values are equal)	{Goldfinger, Ben Hur, Star Wars}	N*ominal Grid* (a region is divided into subregions, in which something can be present or absent)	Colored circle
O	O*rdinal* (can distinguish whether one value is less or greater but not difference or ratio)	⟨Small, Medium, Large⟩	O*rdinal Grid* (order of the subregions is meaningful)	Alpha slider
I	I*nterval* (can do subtraction on values, but no natural zero and can't compute ratios)	[10 Dec. 1978–4 Jun. 1982]	I*nterval Grid* (region has a metric but no distinguished origin)	Year axis
Q	Q*uantitative* (can do arithmetic on values)	[0–100] kg	Q*uantitative Grid* (a region has a metric)	Time slider
Q_s	—*Spatial variables*	[0–20] m	—*Spatial grid*	
Q_m	—*Similarity*	[0–1]	—*Similarity space*	
Q_g	—*Geographical coord.*	[30°N–50°N]Lat.	—*Geographical cood.*	
Q_t	—*Time variable*	[10–20] μsec	—*Time grid*	

TABLE 26.5. Data Table with Meta-Data Describing the Types of the Variables

FilmID	N	230	105	. . .
Title	N	Goldfinger	Ben Hur	. . .
Director	N	Hamilton	Wyler	. . .
Actor	N	Connery	Heston	. . .
Actress	N	Blackman	Harareet	. . .
Year	Q_t	1964	1959	. . .
Length	Q	112	212	. . .
Popularity	Q	7.7	8.2	
Rating	O	PG	G	. . .
FilmType	N	Action	Action	. . .

Source: (Card et al., 1999).

gives importance to spatially encoded variables at the expense of variables encoded using other mappings. Space is perceptually dominant (MacEachren, 1995); it is good for discriminating values and picking out patterns. It is easier, for example, to identify the difference between a sine and a tangent curve when encoded as a sequence of spatial positions than as a sequence of color hues.

Empty space itself, as a container, can be treated as if it had metric structure. Just as we classified variables according to their scale type, we can think of the properties of space in terms of the scale type of an axis of space (cf. Engelhardt, Bruin, Janssen, & Scha, 1996). Axis scale types correspond to the variable scale types (see Table 26.4). The most important axes are

- U = Unstructured (no axis, indicated merely whether something is present or absent)
- N = Nominal Grid (a region is divided into subregions, in which something can be present or absent)
- O = Ordinal Grid (the ordering of these subregions is meaningful), and
- Q = Quantitative Grid (a region has a metric).

Besides these, it is convenient to make additional distinctions for frequently used subtypes, such as Spatial axes (Qs).

Axes can be linear or radial; essentially, they can involve any of the various coordinate systems for describing space. Axes are an important building block for developing Visual Structures. Based on the Data Table for the FilmFinder in Table 26.5, we represent the scatterplot of as composed of two orthogonal quantitative axes:

$$Year \rightarrow Q_x,$$
$$Popularity \rightarrow Q_y.$$

The notation states that the Year variable is mapped to a quantitative X-axis and the Popularity variable is mapped to a quantitative Y-axis. Other axes are used for the FilmFinder query widgets. For example, an ordinal axis is used in the radio buttons for film ratings,

$$Ratings \rightarrow O_y.$$

and a nominal axis is used in the radio buttons for film type,

$$FilmType \rightarrow N_x.$$

Marks. Marks are the visible things that occur in space. There are four elementary types of marks (Fig. 26.11):

1. P = Points (0D),
2. L = Lines (1D),
3. A = Areas (2D), and
4. V = Volumes (3D).

Area marks include surfaces in three dimensions, as well as 2D-bounded regions.

Unlike their mathematical counterpart, point and line marks actually take up space (otherwise, they would be invisible) and may have properties such as shape.

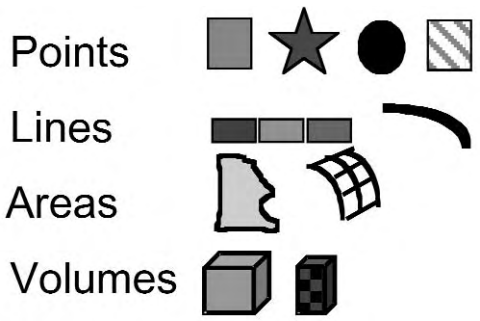


FIGURE 26.11. Types of marks.

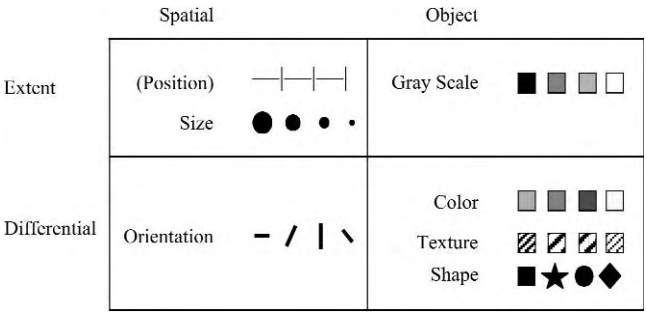


FIGURE 26.12. Retinal properties (Card et al., 1999). The six retinal properties can be grouped by whether they form a scale with a natural zero point (extend) and whether they deal with spatial distance or orientation (spatial).

Connection and enclosure. Point marks and line marks can be used to signify other sorts of topological structure: graphs and trees. These allow showing relations among objects without the geometrical constraints implicit in mapping variables onto spatial axes. Instead, we draw explicit lines. Hierarchies and other relationships can also be encoded using enclosure. Enclosing lines can be drawn around subsets of items. Enclosure can be used for trees, contour maps, and Venn Diagrams.

Retinal properties. Other graphical properties were called retinal properties by Bertin (1967/1983), because the retina of the eye is sensitive to them independent of position. For example, the FilmFinder in Fig. 26.1 uses color to encode information in the scatterplot:

$$FilmID(FilmType) \rightarrow P(Color)$$

This notation says that the FilmType attribute for any FilmID case is visually mapped onto the color of a point.

Figure 26.12 shows Bertin's six "retinal variables" separated into spatial properties and object properties according to which area of the brain they are believed to be processed (Kosslyn, 1994). They are sorted according to whether the property is good for expressing the extent of a scale (has a natural zero point), or whether its principal use is for differentiating marks (Bertin, 1977/1981). Spatial position, discussed earlier as basic visual substrate, is shown in the position it would occupy in this classification.

Other graphical properties have also been proposed for encoding information. MacEachren (1995) has proposed (a) crispness (the inverse of the amount of distance used to blend two areas or a line into an area), (b) resolution (grain with raster or vector data will be displayed), (c) transparency, and (d) arrangement (e.g., different ways of configuring dots). He further proposed dividing color into (a) value (essentially, the gray level of Fig. 26.12), (b) hue, and (c) saturation. Graphical properties from the perception literature that can support preattentive processing have been suggested candidates for coding variables such as curvature, lighting direction, or direction of motion (see Healey, Booth, and Enns, 1995). All of these suggestions require further research.

Temporal encoding. Visual Structures can also temporally encode information; human perception is very sensitive to changes in mark position and the mark's retinal properties. We need to distinguish between temporal data variables to be visualized

$$Q_t \rightarrow some\ visual\ representation$$

and animation, that is, mapping a variable into time,

$$some\ variable \rightarrow Time.$$

Time as animation could encode any type of data (whether it would be an effective encoding is another matter). Time as animation, of course, can be used to visualize time as data.

$$Q_t \rightarrow Time.$$

This is natural, but not always the most effective encoding. Mapping time data into space allows comparisons between two points in time. For example, if we map time and a function of time into space (e.g., time and accumulated rainfall),

$$Q_t \rightarrow Qx\ [make\ time\ be\ the\ X\text{-}axis]$$
$$f(Q_t) \rightarrow Qy,\ [make\ accumulated\ rainfall\ be\ the\ Y\text{-}axis,$$

then we can directly experience rates as visual linear slope, and we can experience changes in rates as curves. This encoding of time into space for display allows us to make much more precise judgments about rates than would be possible from encoding time as time. Another use of time as animation is similar to the unstructured axes of space. Animation can be used to enhance the ability of the user to keep track changes of view or visualization. If the user clicks on some structure, causing it to enlarge and other structures to become smaller, animation can effectively convey the change and the identity of objects across the change, whereas simply viewing the two end states is confusing. Another use is to enhance a visual effect. Rotating a complicated object, for example, will induce 3D effects (hence, allow better reading of some visual mappings).

Expressiveness and Effectiveness

Visual mappings transform Data Tables into Visual Structure and then into a visual image. This image is not just an arbitrary image. It is an image that has a particular meaning it must express. That meaning is the data relation of which it is the visual transformation. We can think of the image as a sentence in a visual language (Mackinlay, 1986b) that expresses the relations in the Data Table. To be a good information visualization, the mappings must satisfy some constraints. The first constraint is that the mapping must be expressive. A visualization is said to be *expressive* if and only if it encodes all the data relations intended and no other data relations. The first part of expressiveness turns out to be easier than the second. Suppose we plot FilmType against Year using the data-to-visual mapping in Fig. 26.13. The problem of this mapping is that the nominal movie rating data are expressed by a quantitative axis. That is, we have tried to map

$$FilmType(N) \rightarrow Position(Q).$$

In so doing, we have visually expressed all the data relation, but the visualization also implies relationships that do not exist. For example, the 1959 version of Ben Hur does not have a film type that is five times greater than the 1926 version of Ben Hur, as implied in the figure. Wisely, the authors of the FilmFinder chose the mapping

$$FilmType(N) \rightarrow Color(N).$$

Of course, there are circumstances in which color could be read as ordinal, or even possibly quantitative, but the miscellaneous order of the buttons in Fig. 26.1 discourages such an interpretation and the relatively low effectiveness of color for this purpose in Table 26.7 also discourages this interpretation.

Table 26.6 shows the mappings chosen by authors of the FilmFinder. The figure shows the Data Table's metadata and data

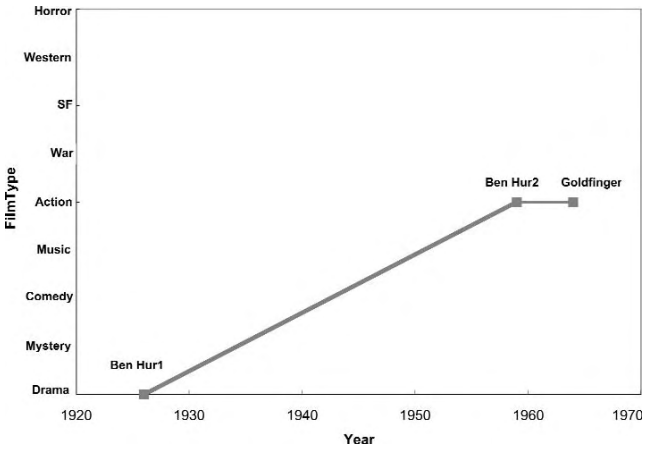


FIGURE 26.13. Mapping from data to visual form that violates expressiveness criterion.

and how they are mapped onto the Visual Structure. Note that the nominal data of the PG ratings is mapped onto a nominal visualization technique (colors). Note also, that names of directors and stars (nominal variables) are raised to ordinal variables (through alphabetization), and then mapped onto an ordinal axis. This is, of course, a common way to handle searching among a large number of nominal items.

Some properties are more effective than others for encoding information. Position is by far the most effective all-around representation. Many properties are more effective for some types of data than for others. Table 26.7 gives an approximate evaluation for the relative effectiveness of some encoding techniques based on (MacEachren, 1995). We note that spatial position is effective for all scale types of data. Shape, on the other hand, is only effective for nominal data. Gray scale is most effective for ordinal data. Such a chart can suggest representations to a visualization designer.

Taxonomy of Information Visualizations

We have shown that the properties of data and visual representation generally constrain the set of mappings that form the basis for information visualizations. Taken together, these constraints form the basis of a taxonomy of information visualizations. Such a taxonomy is given in Table 26.8. Visualizations are grouped into four categories. First are *Simple Visual Structures*, the static mapping of data onto multiple spatial dimensions, trees, or networks plus retinal variables, depicted in Fig. 26.10. Here it is worth distinguishing two cases. There is a perceptual barrier at three (or, in special cases, four) variables, a limit of the amount of data that can be perceived as an immediate whole. Bertin (1977, 1981) called this elementary unit of visual data perception the "image". Although this limit has not been definitively established in information visualization by empirical research, there must be a limit somewhere or else people could simultaneously comprehend a thousand variables. We therefore divide visualizations into those that can be comprehended in an elementary perceptual grasp (three, or in special cases, four variables)—let us call these *direct reading visualizations*—and those more complex than that barrier—which we call *articulated reading visualizations*, in which multiple actions are required.

Beyond the perceptual barrier, direct composition of data relationships in terms of 1, 2, or 3 spatial dimensions plus remaining retinal variables is still possible, but rapidly diminishes in effectiveness. In fact, the main problem of information visualization as a discipline can be seen as devising techniques for accelerating the comprehension of these more complex n-variable data relations. Several classes of techniques for n-variable visualization, which we call *Composed Visual Structures*, are based on composing Simple Visual Structures together by reusing their spatial axes. A third class of Visual Structures—*Interactive Visual Structures*—comes from using the rapid interaction capabilities of the computer. These visualizations invoke the parameter-controlling arrows of Fig. 26.10. Finally, a fourth class of visualizations—*Attention-Reactive Visual Structures*—comes from interactive displays where the system reacts to user actions

TABLE 26.6. Meta-Data and Mappings of Data onto Visual Structure in the FilmFinder

Data							Visual Form				
Variable	Type	Range	$Case_i$	$Case_j$	$Case_k$	…		Type	Visual Structure	Control	Transformation Affected
FilmID	N	All-IDs	230	105	540	…	→	N	Points	Button	All (details)
Title	N	All-titles	Goldfinger	Ben Hur	Ben Hur	…	→sort	O		Alphaslider	Select cases
Director	N	All-directors	Hamilton	Wyler	Niblo	…	→sort	O		Alphaslider	Select cases
Actor	N	All-actors	Connery	Heston	Novarro	…	→sort	O		Alphaslider	Select cases
Actress	N	All-actresses	Blackman	Harareet	McAvoy	…	→sort	O		Alphaslider	Select cases
Year	Q	[1926, 1989]	1964	1959	1926	…	→	Q	X-axis	Axis	Clip range
Length	Q	[0, 450]	112	212	133		→	Q		Two-sided slider	Clip range
Popularity	Q	[1, 9]	7.7	8.2	7.4	…	→	Q	Y-axis	Axis	Clip range
Rating	O	{G, PG, PG-13, R}	PG	G	G	…	→	O		Radio buttons	Select cases
Film Type	N	{Drama, Mystery, Comedy, Music, Action, War, SF, Western, Horror}	Action	Action	Drama		→	N	Color	Radio buttons	Select cases

Source: (Card et al., 1999).

TABLE 26.7. Relative Effectiveness of Position and Retinal Encodings

	Spatial	Q	O	N	Object	Q	O	N
Extent	(Position)	●	●	●	Gray Scale	◐	●	○
	Size	●	●	●	Color	◐	◐	●
Differential	Orientation	◐	◐	●	Texture	◐	◐	●
					Shape	○	○	●

Source: (Card et al., 1999).

by changing the display, even anticipating new displays, to lower to cost of information access and sensemaking to the user. To summarize,

I. Simple Visual Structures
Direct Reading
Articulated Reading
II. Composed Visual Structures
Single-Axis Composition
Double-Axis Composition
Recursive Composition
III. Interactive Visual Structure
IV. Attention-Reactive Visual Structure

These classes of techniques may be combined to produce visualizations that are more complex. To help us keep track of the variable mapping into visual structure, we will use a simple shorthand notation for listing the element of the Visual Structure that the Data Table has mapped into. We will write, for example, [XYR^2] to note that variables map onto the X-axis, the Y-axis, and two retinal encodings. [OX] will indicate that the variables map onto one spatial axis used to arrange the objects (that is, the cases), while another was used to encode the objects' values. Examples of this notation appear in Table 26.8 and Fig. 26.21.

SIMPLE VISUAL STRUCTURES

The design of information visualizations begins with mappings from variables of the Data Table into the Visual Structure. The basic strategy for the visualization designer could be described as follows:

1. *Determine which variables of the Analytic Abstraction to map into spatial position in the Visual Structure.*
2. *Combine these mappings to increase dimensionality (e.g., by folding).*
3. *Use retinal variables as an overlay to add more dimensions.*
4. *Add controls for interaction.*
5. *Consider attention-reactive features to expand space and manage attention.*

We start by considering some of the ways in which variables can be mapped into space.

TABLE 26.8. Taxonomy of Information Visualization Techniques

I. Simple Visual Structures		*III. Interactive Visual Structures*
Direct Reading	Trees	Dynamic queries
1-Variable [X]	Node and link trees	Magic lens
Lists	Enclosure trees	Overview+detail
1D object charts	TreeMaps	Linking and brushing
1D scatterplots	Cone trees	Extraction & comparison
Pie charts	Networks	Attribute xxplorer
Folded Dimensions	Time	
Distributions		*IV. Focus+Context Attention-reactive Visual Abstraction*
Box Plots	*II. Composed Visual Structures*	
2-Variable [XY]		**Data-based Methods**
2D object charts	**Singles-Axis**	Filtering
2D scatterplots	**Composition [XY^n]**	Selective aggregation
3-Variable	Permutation matrices	**View-based methods**
[XYR]	Parallel coordinates	Micro-macro readings
Retinal scatterplot	**Double-Axis**	Highlighting
Kahonen diagrams	**Composition [XY]**	Visual transfer functions
Retinal topographies	Graphs	Perspective distortion
[(XY)Z]	**Recursive Composition**	Alternate geometries
Information landscapes	2D in 2D [$(XY)^{XY}$]	
Information surfaces	Scatterplot matrices	
[XYZ]	Prosection matrices	
3D scatterplots	Hierarchical axes	
4-Variable	Marks in 2D [$(XY)^R$]	
[XYZR]	Stick figures	
3D retinal scatterplots	Color icons	
3D topographies	Shape coding	
	Keim spirals	
—Barrier of Perception—	3D in 3D [$(XYZ)^{XYZ}$]	
Articulated Reading	Worlds within worlds	
n-Variable		
[XYR^{n-2}]		
2D Retinal scatterplots		
[$XYZR^{n-3}$]		
2D Retinal scatterplots		

1-Variable

One-variable visual displays may actually use more than one visual dimension. This is because the data variable or attribute is displayed against some set of objects using some mark and because the mark itself takes space. Or, more subtly, it may be because one of the dimensions is used for arranging the objects and another for encoding via position the variable. A simple example would be when the data are just visually mapped into a simple text list as in Fig. 26.14(a). The objects form a sequence on the Y-dimension, and the width of the marks (the text descriptor) takes space in the X-dimension. By contrast, a one-dimensional scattergraph (Fig. 26.14[b]) does not use a dimension for the objects. Here, the Y-axis is used to display the attribute variable (suppose these are distances from home of gas stations); the objects are encoded in the mark (which takes a little bit of the X-dimension).

More generally, many single-variable visualizations are in the form $v = f(o)$, where v is a variable attribute and o is the object. Figure 26.14(c) is of this form and uses the Y-axis to encode the variable and the X-axis for the objects. Note that if the objects are, as usual, nominal, then they are reorderable: sorting the objects on the variable produces easily perceivable visual patterns. For convenience, we have used rectangular coordinates, but any other orthogonal coordinates could be used as the basis of decomposing space. Figure 26.14(d) uses θ from polar coordinates to encode, say, percentage voting for different presidential candidates. In Fig. 26.14(e), a transformation on the data side has transformed variable o into a variable representing the distribution, then mapped that onto points on the Y-axis. In Fig. 26.14(f), another transformation on the data side has mapped this distribution into 2nd quartiles, 3rd quartiles, and outlier points, which is then mapped on the visual side into a box plot on the Y-axis. Simple as they are, these techniques can be very useful, especially in combination with other techniques.

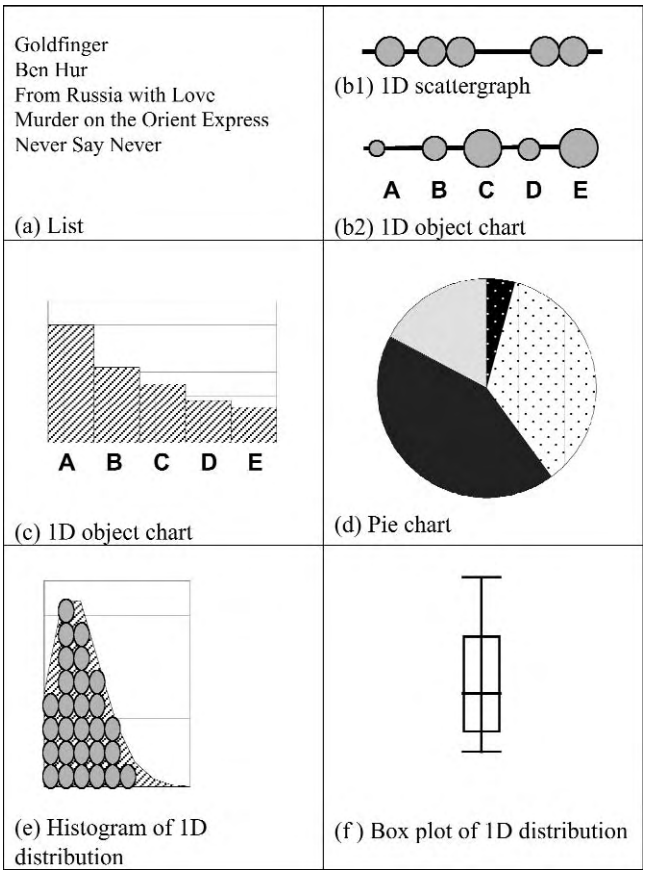


FIGURE 26.14. 1-variable visual abstractions.

One special, but common, problem is how to visualize very large dimensions. This problem occurs for single-variable visualizations, but may also occur for one dimension of a multi-variable visualization. Figure 26.15 shows several techniques for handing the problem. In Fig. 26.15(a) (Freeman & Fertig, 1995), the visual dimension is laid out in perspective. Even though each object may take only one or a few pixels on the axis, the objects are actually fairly large and selectable in the diagram. In Fig. 26.15(b) (Eick, Steffen, & Sumner, 1992), the objects (representing lines of code) are laid out on a *folded* Y-axis. When the Y-axis reaches the bottom of the page, it continues offset at the top. In Fig. 26.15(c) (Keim & Kriegel, 1994), the axis is wrapped in a square spiral. Each object is a single pixel, and its value is coded as the retinal variable color hue. The objects have been sorted on another variable; hence, the rings show the correlation of this attribute with that of the sorting attribute.

One-variable visualizations are also good parts of controls. Controls, in the form of slides, also consume considerable space on the display (for example, the controls in Fig. 26.1) that could be used for additional information communication. Figure 26.15(d) shows a slider on whose surface is a distribution representation of the number of objects for each value of the input variable, thereby communicating information about the slider's sensitivity in different data ranges. The slider on the left of Fig. 26.15(b) has a one-variable visualization that serves as a legend for the main visualization: it associates color hues with dates and allows the selection of date ranges.

2-Variables

As we increase the number of variables, it is apparent that their mappings form a combinatorial design space. Figure 26.16 schematically plots the structure of this space, leaving out the use of multiple lower variable diagrams to plot higher variable combinations. Two-variable visualizations can be thought of as a composition of two elementary axes (Bertin, 1977, 1981; Mackinlay, 1986b), which use a single mark to encode the position on both those axes. Mackinlay called this *mark composition*, and it results in a 2D scattergraph (Fig. 26.16[g]). Note that instead of mapping onto two positional visual encodings, one positional axis could be used for the objects, and the data variables could be mapped onto a position encoding and a retinal encoding (size), as in Fig. 26.16(f).

3-Variables and Information Landscapes

By the time we get to three data variables, a visualization ca be produced in several ways. We can use three separate visual dimensions to encode the three data variables in a *3D scattergraph* (Fig. 26.16[j]). We could also use two spatial dimensions and one retinal variable in a *2D retinal scattergraph* (Fig. 26.16[k]). Or we could use one spatial dimension as an object dimension, one as a data attribute dimension, and one two retinal encodings for the other variables, as in an *object chart such as in* Fig 26.16(i). Because Fig. 26.16(i) uses multiple retinal encodings, however, it may not be as effective as other techniques. Notice that because they all encode three data variables, we have classified 2D and 3D displays together. In fact, one popular 3-variable information visualization that lies between 2D and 3D is the *information landscape* (Fig. 26.16[m]). This is essentially a 2D scattergraph with one data variable extruded into the third spatial dimension. Its essence is that two of the spatial dimensions are more tightly coupled and often relate to a 2D visualization. For example, the two dimensions might form a map with the bars showing the GDP of each region.

Another special type of 3-variable information visualization is a 2D *information topography*. In an information typography, space is partly defined by reference to external structure. For example, the topography of Fig. 26.17(a) is a map of San Francisco, requiring two spatial variables. The size of blue dots indexes the number of domain names registered to San Francisco street addresses. Looking at the patterns in the visualization shows that Internet addresses have especially concentrated in the Mission and South of Mission districts. Figure 26.17(a) uses a topography derived from real geographical space. Various techniques, such as multidimensional scaling, factor analysis, or connectionist self-organizing algorithms, can create abstract spaces based on the similarities among collections of documents or other objects. These abstract similarity spaces can function like a topography. An example can be seen in Fig. 26.17(b), where the pages in a website are depicted as regions in a similarity

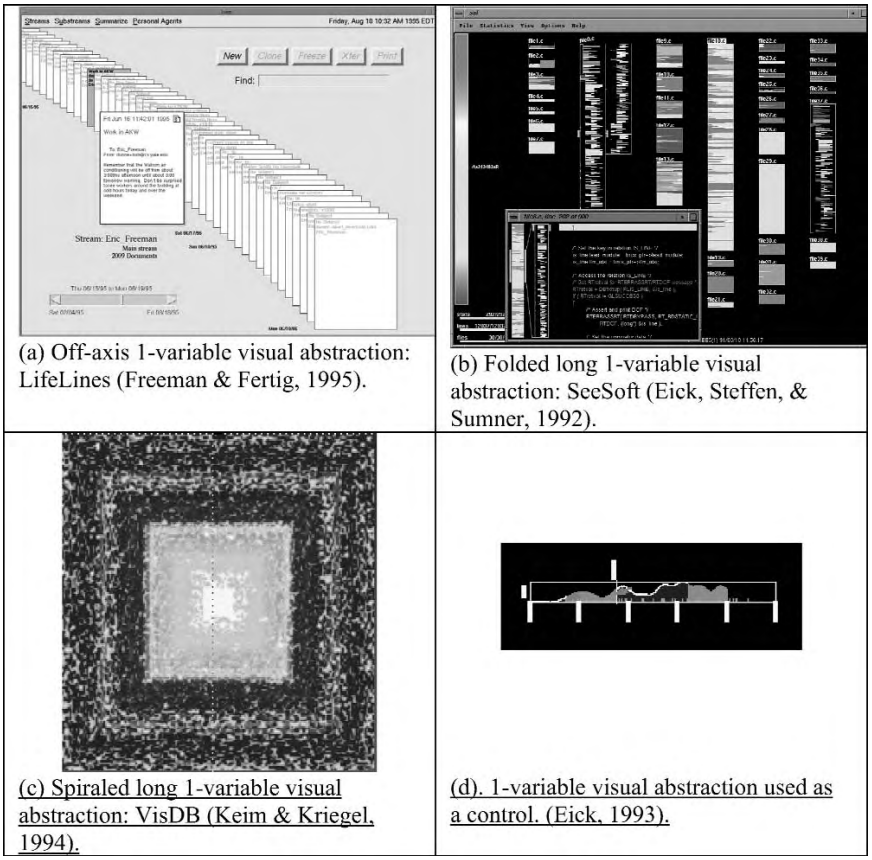


FIGURE 26.15. Uses of 1-variable visual abstractions.

space. To create this diagram[2], a web crawler crawls the site and indexes all the words and pages on the site. Each page is then turned into a document vector to represent the semantic content of that page. The regions are created using a neural network learning algorithm (see Lin, Soergel, & Marchionini (1991)). This algorithm organizes the set of web pages into regions. A visualization algorithm then draws boundaries around the regions, colors them, and names them. The result, called a *Kahonen diagram* after its original inventor, is a type of *retinal similarity topography*.

Information landscapes can also use marks that are surfaces. In Fig. 26.18(a), topics are clustered on a similarity surface, and the strength of each topic is indicated by a 3D contour. A more extreme case is Fig. 26.18(b), where an information landscape is established in spherical coordinates, and the amount of ozone is plotted as a semitransparent overlay on the ρ-axis.

n-Variables

Beyond three variables, direct extensions of the methods we have discussed become less effective. It is possible, of course to make plots using two spatial variables and $n-2$ retinal variables, and the possibilities for four variables are shown in Fig. 26.16. These diagrams can be understood, but at the cost of progressively more effort as the number of variables increases. It would be very difficult to understand an [XYR^{20}] retinal scattergraph, for example.

Trees

An interesting alternative to showing variable values by spatial positioning is to use explicitly drawn linkages of some kind. Trees are the simplest form of these. Trees map cases into subcases. One of the data variables in a Data Table (for example, the variable ReportsTo in an organization chart) is used to define the tree. There are two basic methods for visualizing a tree: (a) Connection and (b) Enclosures.

Connection. Connection uses lines to connect marks signifying the nodes of the tree. Logically, a tree could be drawn merely by drawing lines between objects located randomly positioned on the plane, but such a tree would be visually unreadable. Positioning in space is important. Figure 26.20(a) is a tree from Charles Darwin's notebook (Robin, 1992) drawn to help

[2]This figure is produced by a program called SiteMap by Xia Lin and associates. See http://faculty.cis.drexel.edu/sitemap/index.html.

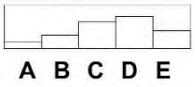
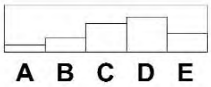
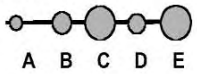
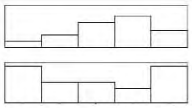
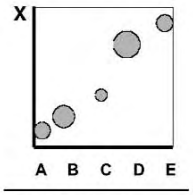
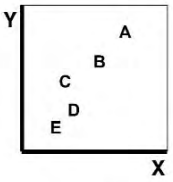
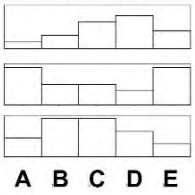
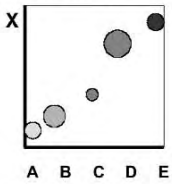
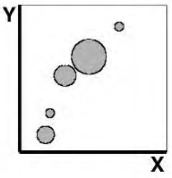
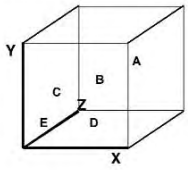
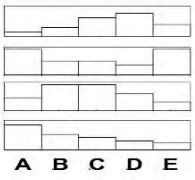
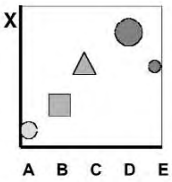
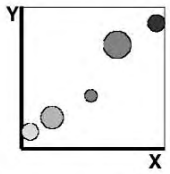
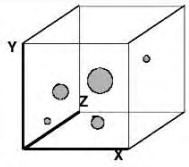
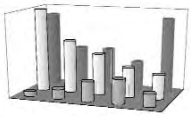
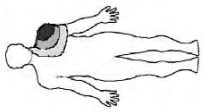
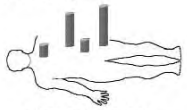
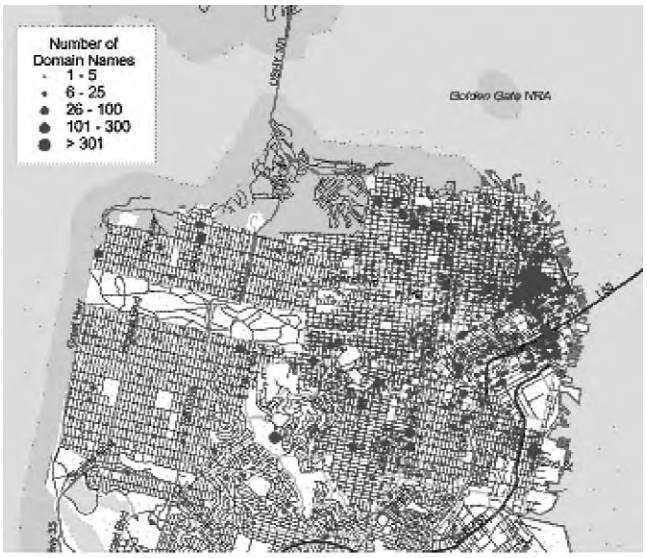
N	Single-Axis Composition	Object Charts	Scatterplots		Topographies
1	 *(a)* [OX] 1D Object chart	 *(b)* [OX] 1D Object chart  *(d)* [OR] 1D Retinal object chart		 *(c)* [X] 1D Scattergraph	
2	 *(e)* [2_OX] Permutation matrix	 *(f)* [[OXR] 2D Object chart		 *(g)* [XY] 2D scattergraph	
3	 *(h)* [3¥OX] Permutation matrix	 *(i)* [OXR^2] 2D Retinal object chart	 *(k)* [XYR] 2D Retinal scattergraph	 *(j)* [XYZ] 3D Scattergraph  *(m)* [(XY)Z] Information Landscape	 *(l)* [X_iY_iR] 2D Retinal topography  *(n)* [$(X_iY_i)R$] Topographic information landscape
4	 *(o)* [4¥OX] Permutation matrix	 *(p)* [OXR^3] 2D Retinal object chart	 *(r)* [XYR^2] 2D Retinal object chart	 *(q)* [XYZR] 3D Retinal scattergraph  *(t)* [(XY)ZR] Retinal information landscape	 *(s)* [XYZR] 3D Retinal topography  *(u)* [(XYZ)R] 3D Topographic information landscape

FIGURE 26.16. Simple Visual Structures..

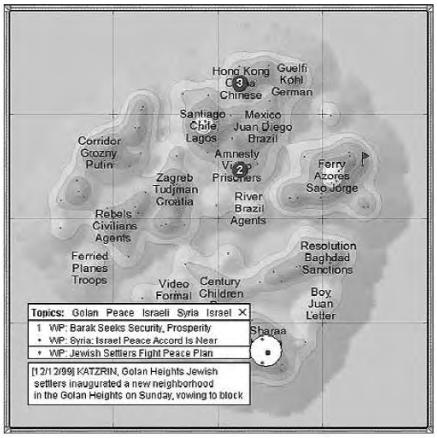


(a) $X_I Y_I R$ Retinal topography

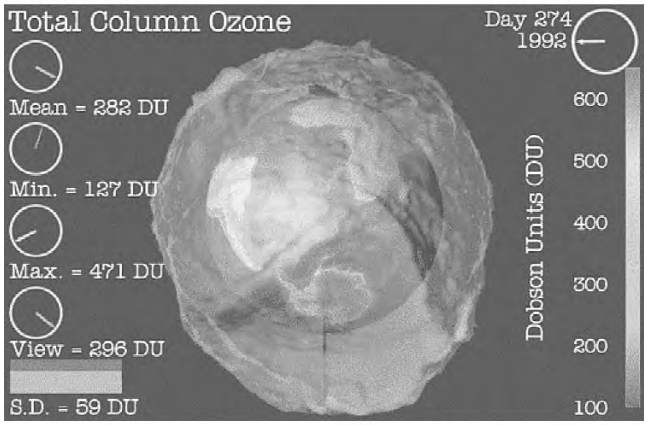


(b) $X_s Y_s R$ Retinal similarity topography

FIGURE 26.17. Retinal information topographies.



(a) News stories based on ThemeScapes (Wise et al., 1995). Courtesy NewsMaps.com.

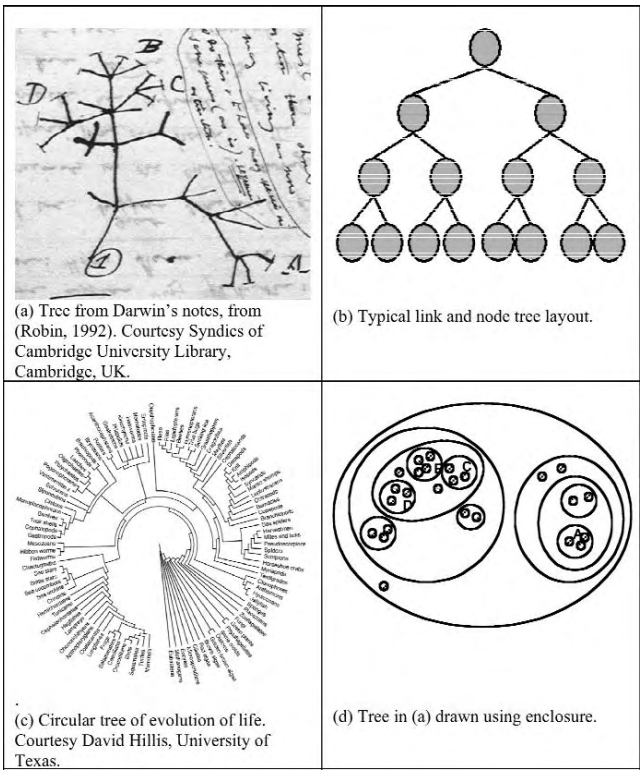


(b) Ozone layer surrounding earth. L. Treinish. Courtesy IBM.

FIGURE 26.18. 3D information surface topographies.

him work out the theory of evolution. Lines proceed from ancestor species to new species. Note that even in this informal setting intended for personal use that the tree uses space systematically (and opportunistically). There are no crossed lines. A common way of laying out trees is to have the depth in the tree map onto one ordinal access as in Fig. 26.20(b), while the other axis is nominal and used to separate nodes. Of course, trees could also be mapped into other coordinate systems: for example, there can be circular trees in which the *r*-axis represents depth and the θ-axis is used to separate nodes as in the representation of the evolution species in Fig. 26.20(c).[3] It is because trees have no cycles that one of the spatial dimensions can be used to encode tree depth. This partial correlation of tree structure and space makes trees relatively easy to lay out and interpret, compared to generalized networks. Hierarchical displays are important not only because many interesting collections of information, such as organization charts or taxonomies, are hierarchical data, but also because important collections of information, such as websites, are *approximately* hierarchical. Whereas practical methods exist for displaying trees up to several thousand nodes, no good methods exist for displaying general graphs of this size. If a visualization problem involves the displaying of network data, a practical design heuristic is to see whether the data might not be forced into a display as a modified tree, such as a tree with a few non-tree links. A significant disadvantage of trees is that as they get large, they acquire an extreme aspect ratio, because the nodes expand exponentially as a function with depth. Consequently, any sufficiently large tree (say, >1000 nodes) resembles a straight line. Circular trees such as Fig. 26.20(c) are one way of trying to buy more space to mitigate this problem. Another disadvantage of trees is the significant empty space between nodes to make their organization easily readable. Various tricks can be used to

[3]This figure is from David Hillis, University of Texas.



(a) Tree from Darwin's notes, from (Robin, 1992). Courtesy Syndics of Cambridge University Library, Cambridge, UK.

(b) Typical link and node tree layout.

(c) Circular tree of evolution of life. Courtesy David Hillis, University of Texas.

(d) Tree in (a) drawn using enclosure.

FIGURE 26.19. Trees.

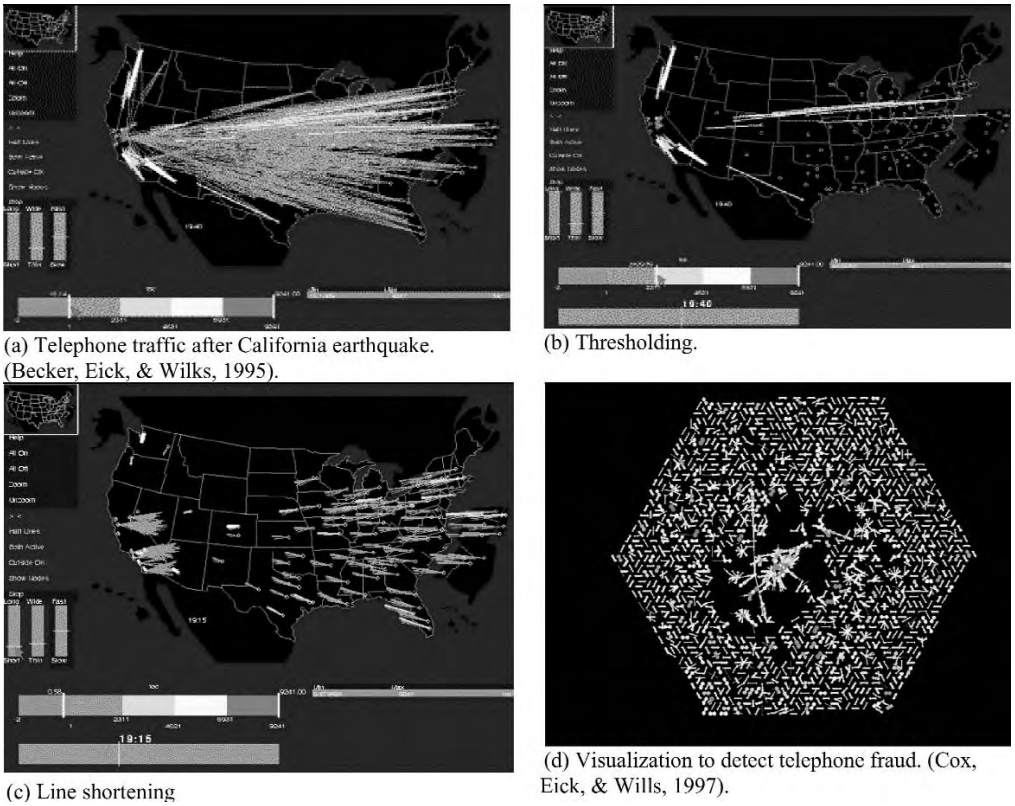
wrap parts of the tree into this empty space, but at the expense of the tree's virtues of readability.

Enclosure. Enclosure uses lines to hierarchically enclose nested subsets of the tree. Figure 26.20(d) is an enclosure tree encoding of Darwin's tree in Fig. 26.20(a). We have already seen one attempt to use tree enclosure, TreeMaps (Fig. 26.5). TreeMaps make use of all the space and stays within prescribed space boundaries, but they do not represent the nonterminal nodes of the tree very well and similar leaves can have wildly different aspect ratios. Recent variations on TreeMaps found ways to "squarify" nodes (Shneiderman & Wattenberg, 2001), mitigating this problem.

Networks

Networks are more general than trees and may contain cycles. Networks may have directional links. They are useful for describing communication relationships among people, traffic in a telephone network, and the organization of the Internet. Containment is difficult to use as a visual encoding for network relationships, so most networks are laid out as node and link diagrams. Unfortunately, straightforward layouts of large node and link diagrams tend to resemble a large wad of tangled string.

We can distinguish the same types of nodes and links in network Visual Structures that we did for spatial axes: (a) Unstructured (unlabeled), (b) Nominal (labeled), (c) Ordinal (labeled



(a) Telephone traffic after California earthquake. (Becker, Eick, & Wilks, 1995).

(b) Thresholding.

(c) Line shortening

(d) Visualization to detect telephone fraud. (Cox, Eick, & Wills, 1997).

FIGURE 26.20. Network methods.

with an ordinal quantity), or (d) Quantitative (weighted links). Retinal properties, such as size or color, can be used to encode information about links and nodes. As in the case of trees, spatial positioning of the nodes is extremely important. Network visualizations escape from the strong spatial constraints of simple Visual Structures only to encounter another set of strong spatial constraints of node links crossing and routing. Networks and trees are not so much an alternative of direct of the direct graphical mappings we have discussed so far as they are another set of techniques that can be overlaid on these mappings. Small node and link diagrams can be laid out opportunistically by hand or by using graph drawing algorithms that have been developed (Battista, Eades, Tamassia, & Tollis, 1994; Cruz & Tamassia, 1998; Tamassia, 1996) to optimize minimal link crossing, symmetry, and other aesthetic principles.

For very large node and link diagrams, additional organizing principles are needed. If there is an external topographic structure, it is sometimes possible to use the spatial variables associated with the nodes. Figure 26.20(a) shows a network based on call traffic between cities in the United States (Becker, Eick, & Wilks, 1995). The geographical location of the cities is used to lay out the nodes of the network. Another way to position nodes is by associating nodes with positions in a similarity space, such the nodes that have the strongest linkages to each other are closest together. There are several methods for computing node nearness in this way. One is to use multidimensional scaling (MDS) (Fairchild, Poltrock, & Furnas, 1988). Another is to use a "spring" technique, in which each link is associated with a Hooke's Law spring weighted by strength of association and the system of springs is solved to obtain node position. Eick and Willis (1993) have argued that the MDS technique places too much emphasis on smaller links. They have derived an alternative that gives clumpier (and hence, more visually structured) clusters of nodes. If positioning of nodes corresponds perfectly with linkage information, then the links do not add more visual information. If positioning does not correspond at all with linkage information, then the diagram is random and obscure. In large graphs, node positions must have a partially correlated relationship to linkage in order to allow the emergence of visual structure. Note that this is what happens in the telephone traffic diagram Fig. 26.20(a). Cities are positioned by geographical location. Communication might be expected to be higher among closer cities, so the fact that communications is heavy between coasts stands out.

A major problem in a network such as Fig. 26.20(a) is that links may obscure the structure of the graph. One solution is to route the links so that they do not obscure each other. The links could even be drawn outside the plane in the third dimension; however, there are limits to the effectiveness of this technique. Another solution is to use *thresholding*, as in Fig. 26.20(b). Only those links representing traffic greater than a certain threshold are included; the others are elided allowing us to see the most important structure. Another technique is *line shortening*, as in Fig. 26.20(c). Only the portion of the line near the nodes is drawn. At the cost of giving up the precise linkage, it is possible to read the density of linkages for the different nodes. Figure 26.20(d) is a technique used to find patterns in an extremely large network. Telephone subscribers are represented as nodes on a hexagonal array. Frequent pairs are located near each other on the array. Suspicious patterns are visible because of the sparseness of the network.

The insightful display of large networks is difficult enough that many information visualization techniques depend on interactivity. One important technique, for example, is node aggregation. Nodes can be aggregated to reduce the number of links that have to be drawn on the screen. Which nodes are aggregated can depend on the portion of the network on which the user is drilling down. Similarly, the sets of nodes can be interactively restricted (e.g., telephone calls greater than a certain volume) to reduce the visualization problem to one within the capability of current techniques.

COMPOSED VISUAL STRUCTURES

So far, we have discussed simple mappings from data into spatial position axes, connections and enclosures, and retinal variables. These methods begin to run into a barrier around three variables as the spatial dimensions are used up and as multiple of the less efficient retinal variables needed. Most interesting problems involve many variables. We shall therefore look at a class of methods that reuse precious spatial axes to encode variables. This is done by composing a compound Visual Structure out of several simple Visual Structures. We will consider five subclasses of such composition: (a) mark composition, (b) case composition, (c) single-axis composition, (d) double-axis composition, and (e) recursive composition. Schematically, we illustrate these possibilities in Fig. 26.21.

Single-axis composition. In single-axis composition, multiple variables that share a single axis are aligned using that axis, as illustrated in Fig. 26.21(a). An example of single-axis composition is a method due to Bertin called *permutation matrices* (Bertin, 1977/1981). In a permutation matrix (Fig. 26.16[o], for example), one of the spatial axes is used to represent the cases and the other a series of bar charts (or rows of circles of different size or some other depiction of the value of each variable) to represent the values. In addition, bars for values below average may be given a different color, as in Fig. 26.7, in order to enhance the visual patterns. The order of the objects and the order of the variables may both be permuted until patterns come into play. Permutation matrices were used in our hotel analysis example. They give up direct reading of the data space in order to handle a larger number of variables. Of course, as the number of variables (or objects) increases, manipulation of the matrices becomes more time-consuming and visual interpretation more complex. Still, permutation matrices or their variants are one of the most practical ways of representing multi-variable data.

If we superimpose the bar charts of the permutation matrix atop one another, and then replace the bar chart with a line linking together the tops of the bars, we get another method for handling multiple variables by single-axis composition—*parallel coordinates* (Inselberg, 1997; Inselberg & Dimsdale, 1990), as shown in Fig. 26.22. A problem is analyzed in parallel coordinates by interactively restricting the objects displayed (the lines) in order to look at cases with common characteristics. In Fig.

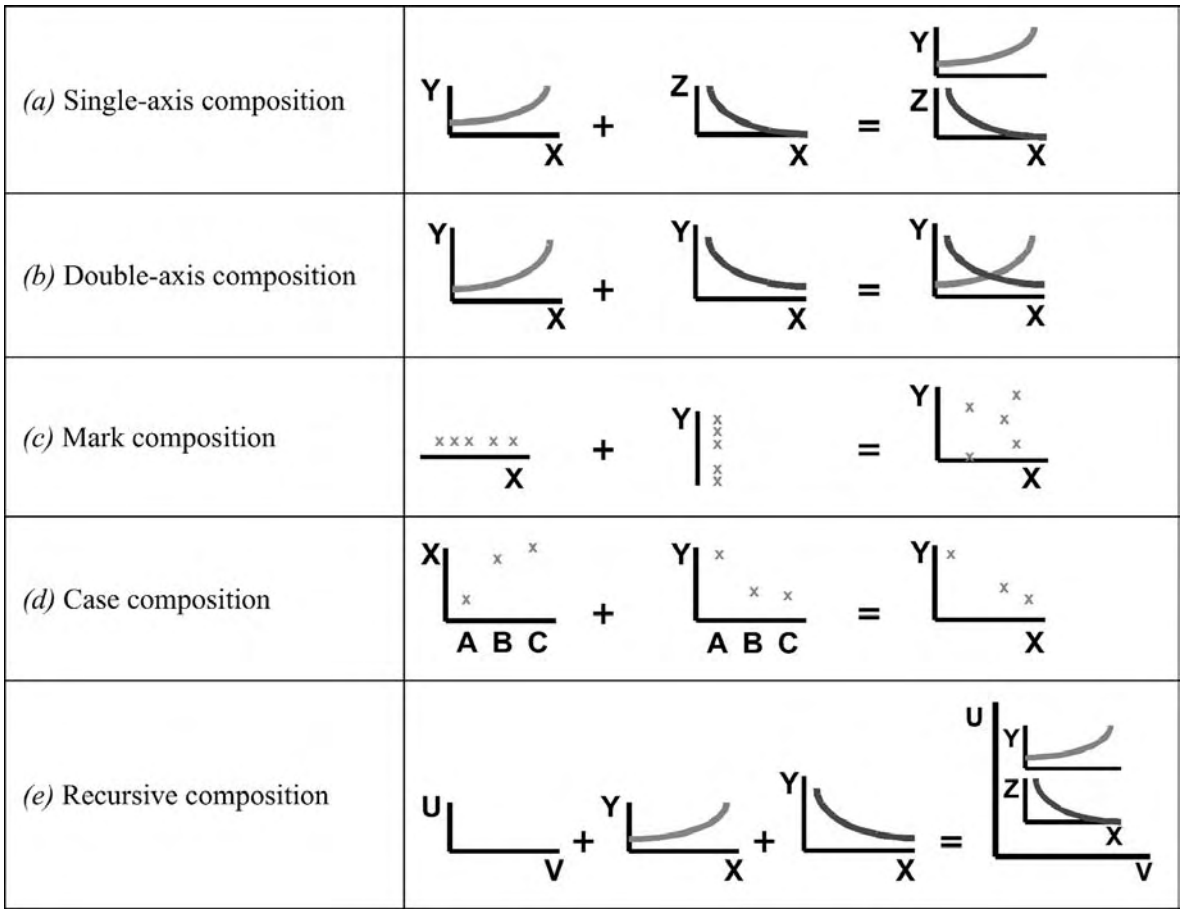


FIGURE 26.21. Composition types.

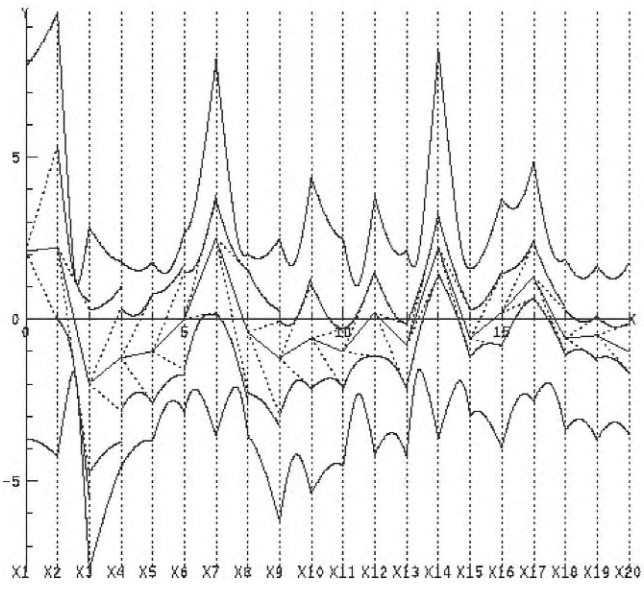


FIGURE 26.22. Single-axis composition: parallel coordinates.

26.22 , parallel coordinates are used to analyze the problem of yield from a certain processor chip. X1 is chip yield, X2 is quality, X3 through X12 are defects, and the rest of the variables are physical parameters. The analysis, looking at those subsets of data with high yield and noticing the distribution of lines on the other parameters, was able to solve a significant problem in chip processing.

Both permutation matrices and parallel coordinates allow analyses in multi-dimensional space, because they are efficient in the use (and reuse) of spatial position and the plane. Actually, they also derive part of their power from being interactive. In the case of permutation matrices, interactivity comes in reordering the matrices. In the case of parallel coordinates, interactivity comes in selecting subsets of cases to display.

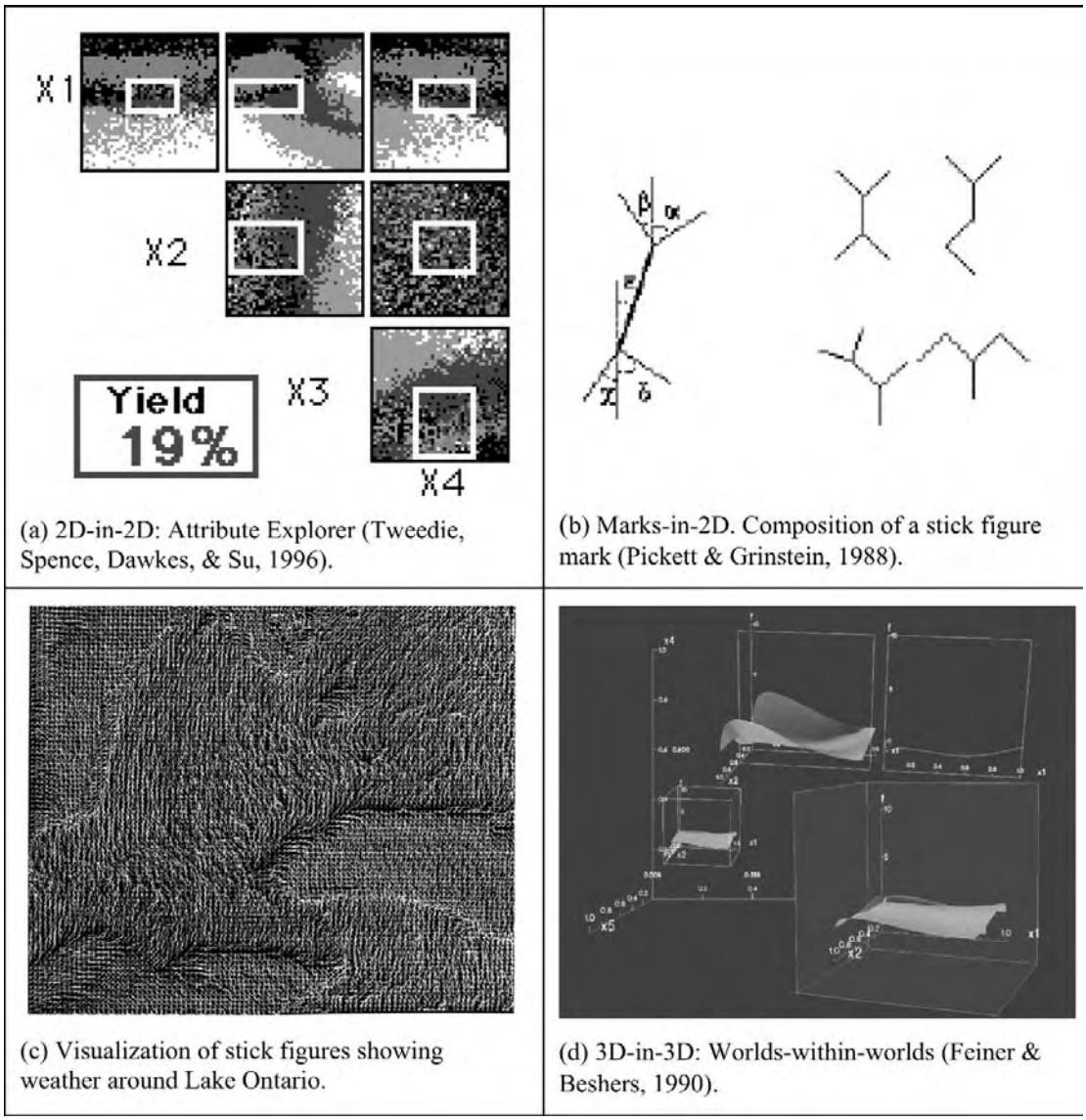
Double-axis composition. In double-axis composition, two visual axes must be in correspondence, in which case the cases are plotted on the same axes as a multivariable graph (Fig. 26.21[b]). Care must be taken that the variables are plotted on a comparable scale. For this reason, the separate scales of the variables are often transformed to a common proportion change scale. An example would be change in price for various stocks.

The cases would be the years, and the variables would be the different stocks.

Mark composition and case composition. Composition can also fuse diagrams. We discussed that each dimension of visual space can be said to have properties as summarized in Table 26.4. The visual space of a diagram is composed from the properties of its axis. In *mark composition* (Fig. 26.21[c]), the mark on one axis can fuse with the corresponding mark on another axis to form a single mark in the space formed by the two axes. Similarly, two object charts can be fused into a single diagram by having a single mark for each case. We call this latter form *case composition* Fig. 26.21(d).

Recursive composition. Recursive composition divides the plane (or 3D space) into regions, placing a subvisualization in each region (Fig. 26.21[e]). We use the term somewhat loosely, since regions have different types of subvisualizations. The FilmFinder in Fig. 26.1 is a good example of a recursive visualization. The screen breaks down into a series of simple Visual Structures and controls: (a) a 3-variable retinal scattergraph (Year, Rating, FilmType) + (b) a 1-variable slider (Title) + (c) a 1-variable slider (Actors) + (d) a 1-variable slider (Actresses) + (e) a 1-variable slider (Director) + (f) a 1-variable slider (FilmLength) + (g) a 1-variable radio button control (Rating) + (h) a 1-variable button-set (FilmType).

Three types of recursive composition deserve special mention: (a)2D-in-2D, (b) marks-in-2D, and (c) 3D-in-3D. An example of *2D-in-2D composition* is the "prosection matrix" (Tweedie, Spence, Dawkes, & Su, 1996) shown in Fig. 26.23(a). Each smaller square in the prosection matrix represents a pair of parameters plotted against each other. The coloring shows



(a) 2D-in-2D: Attribute Explorer (Tweedie, Spence, Dawkes, & Su, 1996).

(b) Marks-in-2D. Composition of a stick figure mark (Pickett & Grinstein, 1988).

(c) Visualization of stick figures showing weather around Lake Ontario.

(d) 3D-in-3D: Worlds-within-worlds (Feiner & Beshers, 1990).

FIGURE 26.23. Recursive composition.

which values of the plotted pair give excellent (red region) or partly good (gray regions) performance for the design of some device. The arrangement of the individual matrices into a supermatrix redefines the spatial dimensions (that is, associates it with different variables) within each of the cells, and the cells themselves are arranged in an overall scheme that systematically uses space. In this way, the precious spatial dimension is effectively expanded to where all the variables can reuse it. An important property of techniques similar to this one is that space is defined at more than one *grain size*, and these levels of grain become the basis for a *macro-micro reading.*

An example of *marks-in-2D composition* in the use of "stick figure" displays. This is an unusual type of visualization in which the recursion is within the mark instead of within the use of space. Figure 26.23(b) shows a mark that is itself composed of submarks. The mark is a line segment with four smaller line segments protruding from the ends. Four variables are mapped onto angle of these smaller line segments and a fifth onto the angle of the main line segment. Two additional variables are mapped onto the position of this mark in a 2D display. A typical result is the visualization in Fig. 26.23(c), which shows five weather variables around Lake Ontario, the outline of which clearly appears in the figure.

Feiner and Beshers (1990) provided an example of the third recursive composition technique, *3D-in-3D composition*. Suppose a dependent variable is a function of six continuous variables, $y = f(x, y, z, w, r, s)$. Three of these variables are mapped onto a 3D coordinate system. A position is chosen in that space, say, $x1, y1, z1$. At that position, a new 3D coordinate system is presented with a surface defined by the other three variables (Fig. 26.23[d]). The user can thus view $y = f(x1, y1, z1, w, r, s)$. The user can slide the second-order coordinate system to any location in the first, causing the surface to change appropriately. Note that this technique combines a composed visual interaction with interactivity on the composition. Multiple second-order coordinate systems can be displayed at the space simultaneously, as long as they do not overlap by much.

INTERACTIVE VISUAL STRUCTURES

In the examples we have considered so far, we have often seen that information visualization techniques were enhanced by being interactive. Interactivity is what makes visualization a new medium, separating it from generations of excellent work on scientific diagrams and data graphics. Interactivity means controlling the parameters in the visualization reference model (Fig. 26.10). This naturally means that there are different types of interactivity, because the user could control the parameters to data transformations, to visual mappings, or to view transformations. It also means that there are different forms of interactivity based on the response cycle of the interaction. As an approximation, we can think of there being three time constants that govern interactivity, which we take to be 0.1 sec, 1 sec, and 10 sec (Card, Moran, & Newell, 1986) (although the ideal value of these may be somewhat less, say, 0.07 sec, 0.7 sec, and 7 sec). The first time constant is the time in which a system response must be made, if the user is to feel that there is a direct physical manipulation of the visualization. If the user clicks on a button or moves a slider, the system needs to update the display in less than 0.1 sec. Animation frames need to take less than 0.1 sec. The second time constant, 1 sec, is the time to complete an immediate action, for example, an animated sequence such as zooming in to the data or rotating a tree branch. The third time constant 10 sec (meaning somewhere in the 5 to 30 sec interval) is the time for completing some cognitive action, for example deleting an element from the display. Let us consider a few well-known techniques for interactive information visualizations.

Dynamic queries. A general paradigm for visualization interaction is dynamic queries, the interaction technique used by the FilmFinder in Fig. 26.1. The user has a visualization of the data and a set of controls, such as sliders, by which subsets of the Data Table can be selected. For example, Table 26.9 shows the mappings of the Data Table and controls for the FilmFinder. The sliders and other controls will select which subset of the data is going to be displayed. In the FilmFinder, the control for Length is a two-sided slider. Setting one end to 90 minutes and the other end to 120 minutes will select for display only those cases of the Data Table whose year variable lies between these limits. The display needs to change within the 0.1 sec of changing the slider.

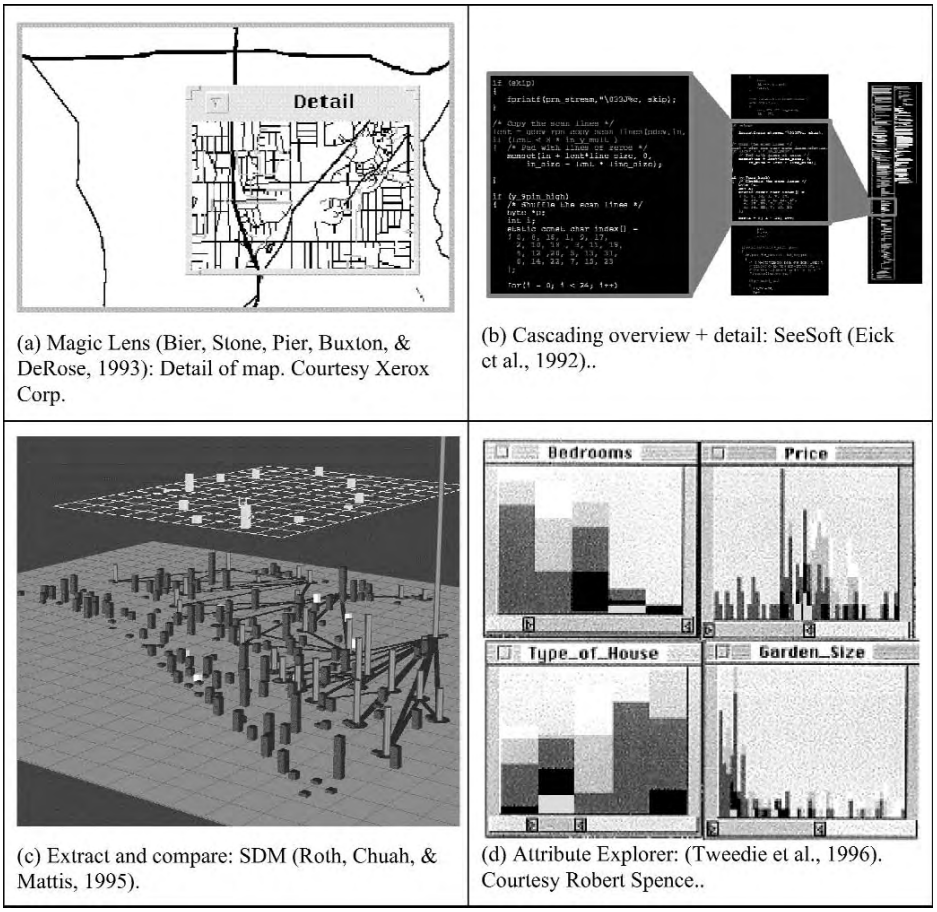
Magic lens (movable filter). Dynamic queries is one type of interactive filter. Another type is a movable filter that can be moved across the display, as in Fig. 26.24(a). These *magic lenses* are useful when it is desired to filter only some of the display. For example, a magic lens could be used with a map that showed the population of any city it was moved over. Multiple magic lenses can be used to cascade filters.

Overview + detail. We can think of an overview + detail display (Fig. 26.24[b]) as a particular type of magic lens, one that magnifies the display and has the magnified region off to the side so as not to occlude the region. Displays have information at different grain sizes. A GIS map may have information at the level of a continent as well as at the level of a city. If the shape of the continent can be seen, the display is too coarse to see the roadways of a city. Overview + detail displays show that data at more than one level, but they also show where the finer grain display fits into the larger grain display. In Fig. 26.24(b), from SeeSoft (Eick et al., 1992), a system for visualizing large software systems, the amount of magnification in the detail view is large enough that two concatenated overview + detail displays are required. Overview + detail displays are thus very helpful for data navigation. Their main disadvantage is that the require coordination of two visual domains.

Linking and brushing. Overview + detail is an example of coordinating dual representations of the same data. These can be coordinated interactively with *linking and brushing.* Suppose, for example, we wish to show power consumption on an airplane, both in terms of the physical representation of the airplane and a logical circuit diagram. The two views could be shown and *linked* by using the same color for the same component types. Interactivity itself can be used for a dynamic form of linking called *brushing*. In brushing, running the

TABLE 26.9. Visual Marks and Controls for FilmFinder

Data							Visual Form				
Variable	Type	Range	$Case_i$	$Case_j$	$Case_k$	...		Type	Visual Structure	Control	Transformation Affected
FilmID	N	All-IDs	230	105	540	...	→	N	Points	Button	All (details)
Title	N	All-titles	Goldfinger	Ben Hur	Ben Hur	...	→sort	O		AlphaSlider	Select cases
Director	N	All-directors	Hamilton	Wyler	Niblo	...	→sort	O		AlphaSlider	Select cases
Actor	N	All-actors	Connery	Heston	Novarro	...	→sort	O		AlphaSlider	Select cases
Actress	N	All-actresses	Blackman	Harareet	McAvoy	...	→sort	O		AlphaSlider	Select cases
Year	Q	[1926, 1989]	1964	1959	1926	...	→	Q	X-axis	Axis	Clip range
Length	Q	[0, 450]	112	212	133		→	Q		Two-sided slider	Clip range
Popularity	Q	[1, 9]	7.7	8.2	7.4	...	→	Q	Y-axis	Axis	Clip range
Rating	O	{G, PG, PG-13, R}	PG	G	G	...	→	O		Radio buttons	Select cases
Film Type	N	{Drama, Mystery, Comedy, Music, Action, War, SF, Western, Horror}	Action	Action	Drama		→	N	Color	Radio buttons	Select cases



(a) Magic Lens (Bier, Stone, Pier, Buxton, & DeRose, 1993): Detail of map. Courtesy Xerox Corp.

(b) Cascading overview + detail: SeeSoft (Eick et al., 1992)..

(c) Extract and compare: SDM (Roth, Chuah, & Mattis, 1995).

(d) Attribute Explorer: (Tweedie et al., 1996). Courtesy Robert Spence..

FIGURE 26.24. Interaction techniques.

cursor over a part of one of the views causes highlighting both in that view and in the other view.

Extraction and comparison. We can also use interaction to extract a subset of the data to compare with another subset. An example of this is in the SDM system (Chuah, Roth, Mattis, & Kolojejchick, 1995) in Fig. 26.24(c). The data are displayed in a 3D information landscape, but the perspective interferes with the ability to compare it. Information is therefore *extracted* from the display (leaving ghosts behind) and placed in an orthogonal viewing position where it can be *compared* using 2D. It could also be dropped into another display. Interactivity makes possible these manipulations, while keeping them coordinated with the original representations.

Attribute explorer. Several of these interactive techniques are combined in the Attribute Explorer (Tweedie et al., 1996). Figure 26.24(d) shows information on four attributes of houses. Each attribute is displayed by a histogram, where each square making up the histogram represents an individual house. The user selects a range of some attribute, say price. Those pixels making up the histogram on price have their corresponding pixels linked representing houses highlighted on the other attributes. Those houses meeting all the criteria are highlighted in one color; those houses meeting, say, all but one are highlighted in another color. In this way, the user can tell about the "near misses." If the users were to relax one of the criteria only a little (say, reducing price by $100), then the user might be able to gain more on another criterion (say, reducing a commute by 20 miles).

FOCUS + CONTEXT ATTENTION-REACTIVE ABSTRACTIONS

So far, we have considered visualizations that are static mappings from Data Table to Visual Structure and those where the mappings Data Table to Visual Structure are interactively controlled by the user. We now consider visualizations in which the machine is no longer passive, but its mappings from Visual Structure to View are altered by the computer according to the its model of the user's *degree of interest*. We can, in principle, associate a cost of access with every element in the Data Table. Take the FilmFinder in Figure 26.3. Details about the movie "Murder on the Orient Express" are accessible at low cost in terms of time because they are presently visible on the screen. Details of "Goldfinger," a movie with only a mark on the display, take more time to find. Details of "Last Year at Marienbad," a movie with no mark on the display, would take much more time. The idea is that with a model for predicting users' changes in interest, the system can adjust its displays to make costs lower for information access. For example, if the user wants some detail about a movie, such as the director, the system can anticipate that the user is more likely to want other details about the movie as well and therefore display them all at the same time: The user does not have execute a separate command; the cost is therefore reduced.

Focus+context views are based on several premises: First, the user needs both overview (context) and detail information (focus) during information access, and providing these in separate screens or separate displays is likely to cost more in user time. Second, information needed in the overview may be different from that needed in the detail. The information of the overview needs to provide enough information the user to decide where to examine next or to give a context to the detailed information rather than the detailed information itself. As Furnas (1981) has argued, the user's interest in detail seems to fall away in a systematic way with distance as information objects become farther from current interest. Third, these two types of information can be combined within a single dynamic display, much as human vision uses a two-level focus and context strategy. Information broken into multiple displays (separate legends for a graph, for example) seem to degrade performance due to reasons of visual search and working memory.

Furnas (1981) was the first to articulate these ideas systematically in his theory of *fisheye views*. The essence of focus+context displays is that the average cost of accessing information is reduced by placing the most likely needed information for navigation and detail where it is fastest to access. This can be accomplished by working on either the data side or the visual side of the visual reference model, Fig. 26.10. We now consider these techniques in more detail.

Data-Based Methods

Filtering. On the data side, focus+context effects can be achieved by filtering out which items from the Data Table are actually displayed on the screen. Suppose we have a tree of categories taken from Roget's Thesaurus, and we are interacting with one of these, "Hardness."

Matter
  ORGANIC
    Vitality
      Vitality in general
      Specific vitality
    Sensation
      Sensation in general
      Specific sensation
  INORGANIC
    Solid
      Hardness
      Softness
    Fluid
      Fluids in general
      Specific fluids

Of course, this is a small example for illustration. A tree representing a program listing or a computer directory or a taxonomy could easily have thousands of lines, a number that would vastly exceed what could fit on the display and hence would have a high cost of accessing. We calculate a degree-of-interest (DOI) for each item of the tree, given that the focus is on the node Hardness. To do this, we split the DOI into an intrinsic

part and a part that varies with distance from the current center of interest and use a formula from Furnas (1981).

$$DOI = Intrinsic\ DOI + Distance\ DOI$$

Figure 26.25 shows schematically how to perform this computation for our example. We assume that the intrinsic DOI of a node is just its distance of the root (Fig. 26.25[a]). The distance part of the DOI is just the traversal distance to a node from the current focus node (Fig. 26.25[b]; it turns out to be convenient to use negative numbers for this computation, so that the maximum amount of interest is bounded, but not the minimum amount of interest). We add these two numbers together (Fig. 26.25 [c]) to get the DOI of each node in the tree. Then we apply a minimum threshold of interest (−5 in this case) and only show nodes more interesting than that threshold. The result is the reduced tree:

Matter
  INORGANIC
  ORGANIC
    Solid
      Hardness
      Softness
    Fluid

The reduced tree gives local context around the focus node and progressively less detail farther away. But it does seem to give the important context.

Selective aggregation. Another focus+context technique from the data side is selective aggregation. Selective aggregation creates new cases in the Data Table that are aggregates of other cases. For example, in a visualization of voting behavior in a presidential election, voters could be broken down by sex, precinct, income, and party affiliation. As the user drills down on, say, male Democrats earning between $25,000 and $50,000, other categories could be aggregated, providing screen space and contextual reference for the categories of immediate interest.

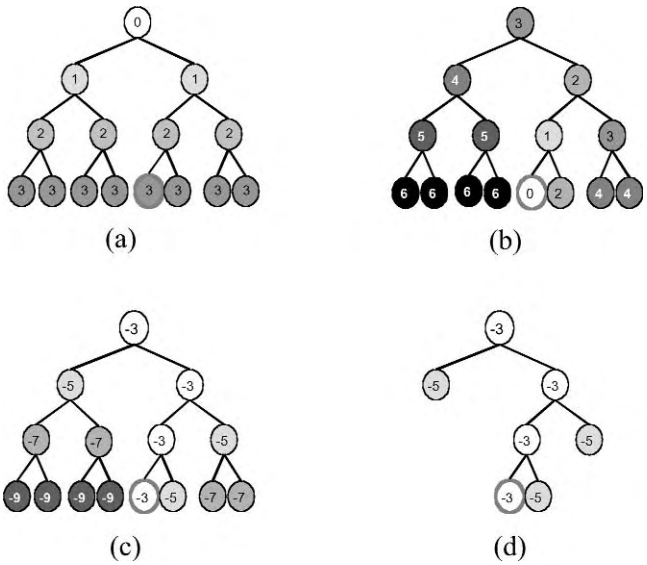


FIGURE 26.25. Degree-of-Interest calculation for fish-eye visualization.

View-Based Methods

Micro-macro readings. Micro-macro readings are diagrams in which "detail cumulates into larger coherent structures" (Tufte, 1990). The diagram can be graphically read at the level of larger contextual structure or at the detail level. An example is Fig. 26.26. The micro reading of this diagram shows three million observations of the sleep (lines), wake (spaces), and feeding (dots) activity of a newborn infant. Each day's activity is repeated three times on a line to make the cyclical aspect of the activity more clearly visible. The macro reading of the diagram, emphasized the thick lines, shows the infant transitioning from the natural human 25-hour cycle at birth to the 24-hour solar day. The macro reading serves as context and index into the micro reading.

Higblighting. Highlighting is a special form of micro-macro reading in which focal items are made visually distinctive in some way. The overall set of items provides a context for the changing focal elements.

Visual transfer functions. We can also warp the view with viewing transformations. An example is a visualization called the *bifocal lens* (Spence & Apperley, 1982). Fig. 26.27(a) shows a set of documents the user would like to view, but which is too large to fit on the screen. In a bifocal lens, documents not in a central focal region are compressed down to a smaller size. This could be a strict visual compression. It could also involve a change in representation. We can talk about the visual compression in terms of a visual transfer function Fig. 26.27(b), sometimes conveniently represented in terms of its first derivative in Fig. 26.27(c). This function shows how many units of an

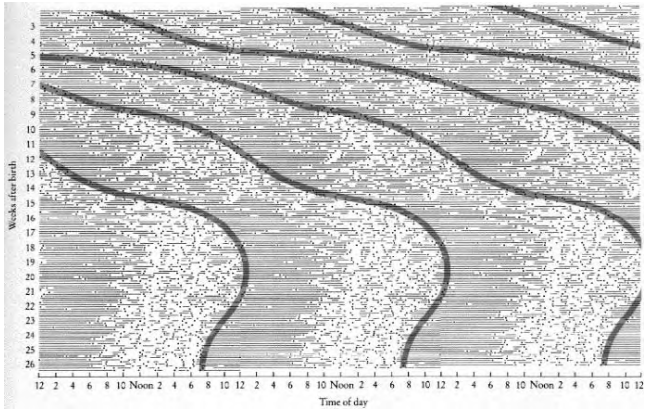


FIGURE 26.26. Micro-macro reading. (Winfree, 1987). Courtesy Scientific American Library.

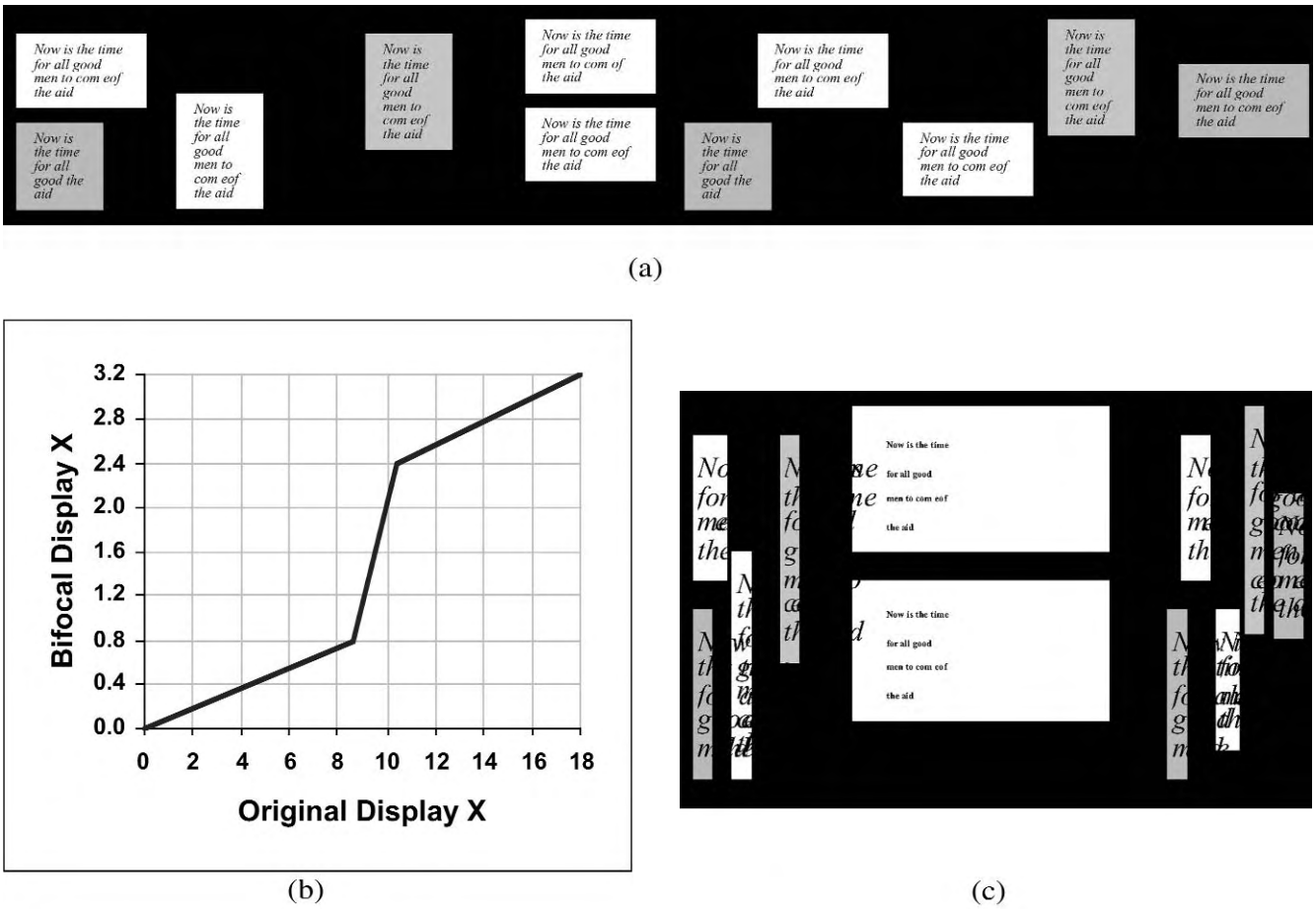
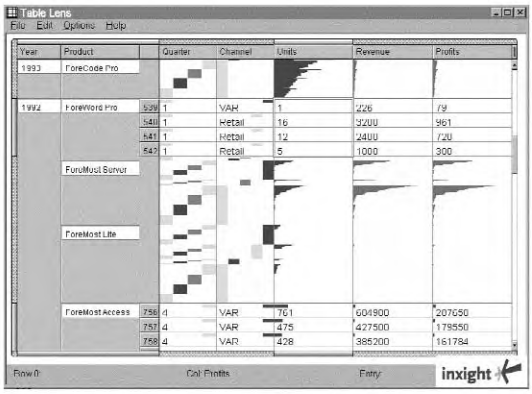


FIGURE 26.27. Bifocal + transfer function.

axis in the original display are mapped into how many units in the resultant display. The result could be compression or enlargement of a section of the display. As a result of applying this visual transfer function to Fig. 26.27(a), the display is compressed to Fig. 26.27(d). Actually, the documents in the compressed region have been further altered by using a semantic zooming function to give them a simplified visual form. The form of Fig. 26.27(c) shows that this is essentially a step function of two different slopes. An example of a two-dimensional step function is the Table Lens (Fig. 26.28[a]). The Table Lens is a spreadsheet in which the columns of selected cells are expanded to full size in X and the rows of selected cells are expanded to full size in Y. All other cells are compressed, and their content represented only by a graphic. As a consequence, spreadsheets up to a couple orders of magnitude larger can be represented.

By varying the visual transfer function (see, for example, the review by Leung and Apperley (1994), a wide variety of distorted views can be generated. Figure 26.28(b) shows an application in which a visual transfer function is used to expand a bubble around a local region on a map. The expanded space in the region is used to show additional information about that region. Distorted views must be designed carefully so as not to damage important visual relationships. Bubble distortions of maps may change whether roads appear parallel to each other. However, distorted views can be designed with "flat" and "transition" regions to address this problem. Figure 26.27(a) does not have curvilinear distortions. Focus+context visualizations can be used as part of compact user controls. Keahey (2001) has created an interactive scheme in which the bubble is used to "preview" a region. When the user releases a button over the region, the system zooms in far enough to flatten out the bubble. Bederson has developed a focus+context pull-down menu (Bederson, 2000) that allows the viewing and selection of large lists of typefaces in text editor Fig. 26.27(c).

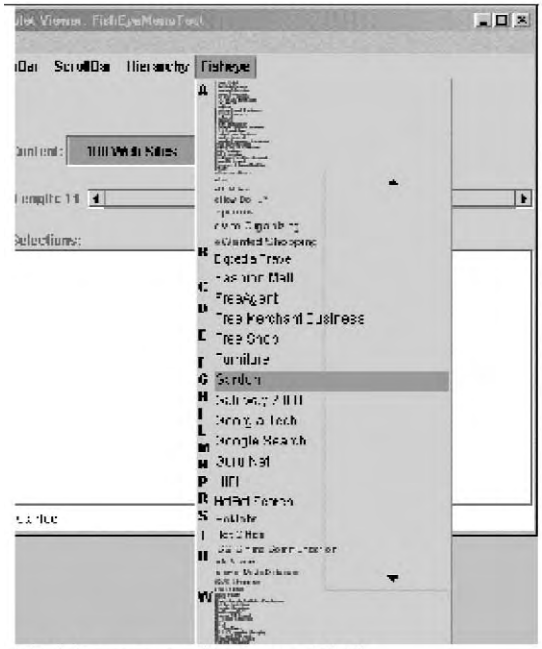
Perspective distortion. One interesting form of distorting visual transfer functions is 3D perspective. Although it can be described with a 2D distorting visual transfer function, it is usually not experienced as distorting by users due to the special perceptual mechanisms humans have for processing 3D. Figure 26.28(c) shows the Perspective Wall (Mackinlay, Robertson, & Card, 1991). Touching any place on the walls animates its transition into the central focal area. The user perceives the



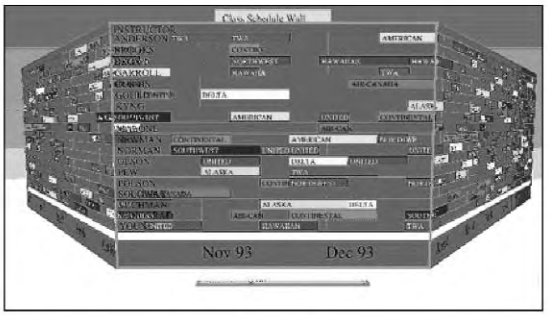
(a) Table Lens. Courtesy of Inxight Software.



(b) Nonlinear distortion of UK.. Courtesy Alan Keahey



(c) Fisheye menus (Bederson, 2000).



(d) Perspective Wall (Mackinlay, Robertson, & Card, 1991).

FIGURE 26.28. Attention-Reactive Visualizations.

context area of the wall as an undistorted 2D image in a 3D space, rather than as a distorted 2D image; however, the same sort of compression is still achieved in the nonfocus area.

Alternate geometries. Instead of altering the size of components, focus+context effects can also be achieved by changing the geometry of the spatial substrate itself. One example is the hyperbolic tree (Lamping & Rao, 1994). A visualization such as a tree is laid out in hyperbolic space (which itself expands exponentially, just like the tree does), and then projected on to the Euclidean plane. The result is that the tree seems to expand around the focal nodes and to be compressed elsewhere. Selecting another node in the tree animates that portion to the focal area. Munzner (Munzner & Burchard, 1995) has extended this notion to 3D hyperbolic trees and used them to visualize portions of the Internet.

SENSEMAKING WITH VISUALIZATION

Knowledge Crystallization

The purpose of information visualization is to amplify cognitive performance, not just to create interesting pictures. Information visualizations should do for the mind what automobiles do

for the feet. So here, we return to the higher level cognitive operations of which information visualization is a means and a component. A recurrent pattern of cognitive activity to which information visualization would be useful (though not the only one!) is "knowledge crystallization." In knowledge crystallization tasks, there is a goal (sometimes ill-structured) that requires the acquisition and making sense of a body of information, as well as the creative formulation of a knowledge product, decision, or action. Examples would be writing a scientific paper, business or military intelligence, weather forecasting, or buying a laptop computer. For these tasks, there is usually a concrete outcome of the task—the submitted manuscript of a paper, a delivered briefing, or a purchase. Knowledge crystallization does have characteristic processes, however, and it is by amplifying these that information visualization seeks to intervene and amplify the user's cognitive powers. Understanding of this process is still tentative, but the basic parts can be outlined:

Acquire information. Make sense of it. Create something new. Act on it.

In Table 26.10, we have listed some of the more detailed activities these entail. We can see examples of these in our initial examples.

Acquire information. The FilmFinder is concentrated largely on acquiring information about films. *Search* is one of the methods of acquiring information in Table 26.10, and the FilmFinder is an instance of the use of information visualization in search. In fact, Shneiderman (Card et al., 1999) has identified a heuristic for designing such systems:

Overview first, zoom and filter, then details-on-demand

The user starts with an overview of the films, and then uses sliders to filter the movies, causing the overview to zoom in on the remaining films. Popping up a box gives details on the particular films. The user could use this system as part of a knowledge crystallization process, but the other activities would take place outside the system. The SmartMoney system also uses the TreeMap visualization for acquiring information, but this time the system is oriented toward *monitoring,* another of the methods in Table 26.10. A glance at the sort of chart in Fig. 26.5 allows an experienced user to notice interesting trends among the hundreds of stocks and industries monitored. Another method of acquiring information, *capture*, refers to acquiring information that is tacit or implicit. For example, when users browse the World Wide Web, their paths contain information about their goals. This information can be captured in logs, analyzed, and visualized (Chi & Card, 1999). It is worth making the point that acquiring information is not something that the user must necessarily do explicitly. Search, monitoring, and capture can be implicitly triggered by the system.

TABLE 26.10. Knowledge Crystallization Operators

Acquire Information	Monitor
	Search,
	Capture (make implicit knowledge explicit)
Make sense of it	Extract information
	Fuse different sources
	Find schema
	Recode information into schema
Create something new	Organize for creation
	Author
Act on it	Distribute
	Apply
	Act

Make sense of it. The heart of knowledge crystallization is sensemaking. This process is by no means as mysterious as it might appear. Because sensemaking involving large amounts of information must be externalized, the costs of finding, organizing, and moving information around have a major impact on its effectiveness. The actions of sensemaking itself can be analyzed. One process is *extraction.* Information must be got out of its sources. In our hotel example, the hotel manager extracted information from hotel records. A more subtle issue is that information from different sources must be *fused*—that is, registered in some common correspondence. If there are six called-in reports of traffic accidents, does this mean six different accidents, one accident called in six times, or two accidents reported by multiple callers? If one report merely gives the county, while another just gives the highway, it may not be easy to tell. Sensemaking involves finding some *schema*—that is, some descriptive language—in terms of which information can be compactly expressed (Russell, Stefik, Pirolli, & Card, 1993). In our hotel example, permuting the matrices brought patterns to the attention of the manager. These patterns formed a schema she used to organize and represent hotel stays compactly. In the case of buying a laptop computer, the schema may be a table of features by models. Having a common schema then permits compact description. Instances are *recoded* into the schema. Residual information that does not fit the schema is noted and can be used to adjust the schema.

Create something new. Using the schema, information can be reorganized to create something new. It must be *organized* into a form suitable for the output product and that product must be *authored.* In the case of the hotel example, the manager created the presentation of Fig. 26.7(c).

Act on it. Finally, there is some consequential output of the knowledge crystallization task. That action may be to distribute a report or give a briefing, to act directly in some way, such as setting up a new promotion program for the hotel or buying a laptop on the basis of the analysis, or by giving directives to an organization.

Levels for Applying Information Visualization

Information visualization can be applied to facilitate the various subprocesses of knowledge crystallization just described. It can also be applied at different architectural levels in a system. These have been depicted in Fig. 26.29. At one level is the use of visualization to help users access information outside the im-

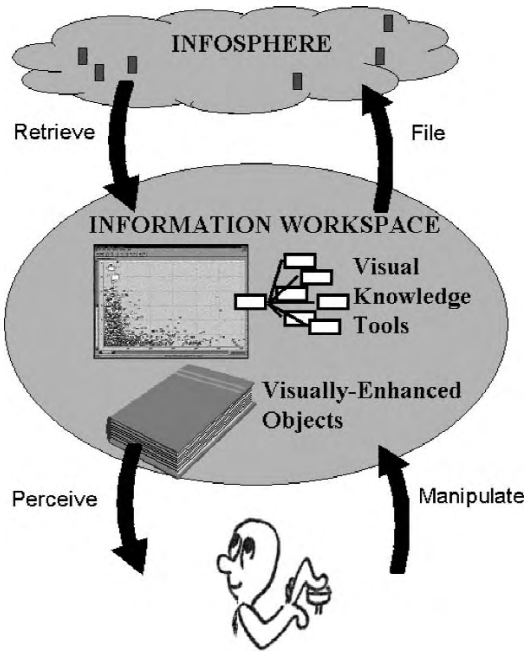
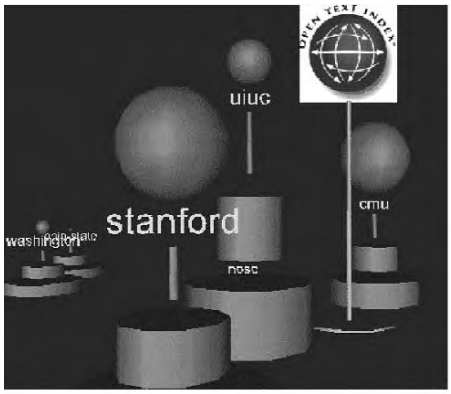


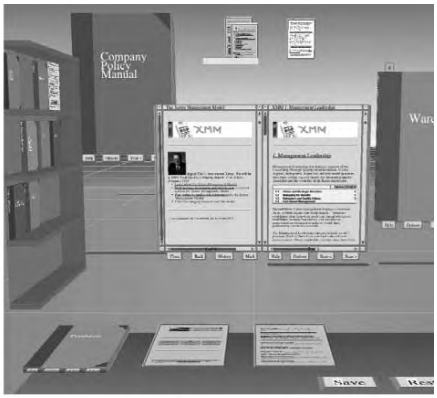
FIGURE 26.29. Levels of use for information visualization.

mediate environment—the *infosphere*—such as information on the Internet or from corporate digital libraries. Figure 26.30(a) shows such a visualization of the Internet (Bray, 1996). websites are laid out in a space such that sites closer to each other in the visualization tend to have more traffic. The size of the disk represents the number of pages in the site. The globe size represents the number of out-links. The globe height shows the number of in-links.

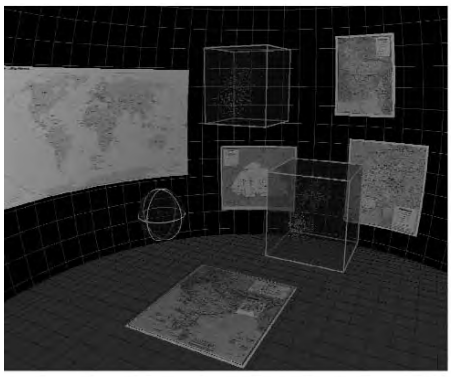
The second level is the *information workspace*. The information workspace is like a desk or workbench. It is a staging area for the integration of information from different sources. An information workspace might contain several visualizations related to one or several tasks. Part of the purpose of an information workspace is to make the cost of access low for information in active use. Figure 26.30(b) shows a 3D workspace for the Internet, the Web Forager (Card, Robertson, & York, 1996). Pages from the World Wide Web, accessed by users through clicking on URLs or searches, appear in the space. These can be organized into piles or books related to different topics. Figure 26.30 (c) shows another document workspace, STARLIGHT (Risch et al., 1997). Documents are represented as galaxies of points in space such that similar documents are near each other. In the workspace, various tools allow linking the documents to maps and other information and analytical resources.



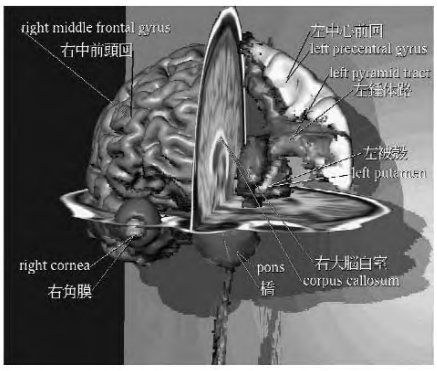
(a) Infosphere: (Bray, 1996).



(b) Workspace: Web Forager (Card, Robertson, & York, 1996).



(c) Workspace: STARLIGHT: (Risch et al., 1997).



(d) Visually-enhanced object: Voxel-Man. Courtesy of University of Hamburg.

FIGURE 26.30. Information visualization applications.

The third level is *visual knowledge tools*. These are tools that allow schema forming and rerepresentation of information. The permutation matrices in Fig. 26.7, the SeeSoft system for analyzing software in Fig. 26.15(b), and the Table Lens in Fig. 26.27(a) are examples of visual knowledge tools. The focus is on determining and extracting the relationships.

The final level is *visually enhanced objects*, coherent information objects enhanced by the addition of information visualization techniques. An example is Fig. 26.30(d), in which voxel data of the brain have been enhanced through automatic surface rendition, coloring, slicing, and labeling. Abstract data structures representing neural projects and anatomical labels have been integrated into a display of the data. Visually enhanced objects focus on revealing more information from some object of intrinsic visual form.

Information visualization is a set of technologies that use visual computing to amplify human cognition with abstract information. The future of this field will depend on the uses to which it is put and how much advantage it gives to these. Information visualization promises to help us speed our understanding and action in a world of increasing information volumes. It is a core part of a new technology of human interfaces to networks of devices, data, and documents.

ACKNOWLEDGMENT

This chapter benefited from detailed and generous discussion with Jock Mackinlay and tutorial collaborations with Ed Chi of PARC.

References

Ahlberg, C., & Shneiderman, B. (1994a). *Visual information seeking using the filmfinder.* Paper presented at the Conference Companion of CHI'94, ACM Conference on Human Factors in Computing Systems.

Ahlberg, C., & Shneiderman, B. (1994b). *Visual information seeking: Tight coupling of dynamic query filters with starfield displays.* Paper presented at the Proceedings of CHI'94, ACM Conference on Human Factors in Computing Systems, New York.

Battista, G. D., Eades, P., Tamassia, R., & Tollis, I. G. (1994). Annotated bibliography on graph drawing. *Computational Geometry: Theory and Applications, 4*(5), 235–282.

Becker, R. A., Eick, S. G., & Wilks, A. R. (1995, March 1). Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics, 1*, 16–28.

Bederson, B. B. (2000). *Fisheye menus.* Paper presented at the UIST 2000, ACM Symposium on User Interface Software and Technology (CHI Letters, 2(2)).

Bertin, J. (1983). *Semiology of graphics: Diagrams, networks, maps* (W. J. Berg, Trans.). Madison, WI: University of Wisconsin Press. (Original work published 1967)

Bertin, J. (1981). Graphic constructions (P. Scott, Trans.). In *Graphics constructions and graphic information-processing* (pp. 24–31). Berlin, Germany: Walter De Gruyter. (Original work published 1977)

Bray, T. (1996). Measuring the web. *Computer Networks and ISDN Systems, 28*(7–11-May), 992.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Information visualization: Using vision to think*. San Francisco: Morgan Kaufmann Publishers.

Card, S. K., Moran, T. P., & Newell, A. (1986). The model human processor: An engineering model of human performance. In J. Thomas (Ed.), *Handbook of perception and human performance* (pp. 41–35). New York: John Wiley and Sons.

Card, S. K., Robertson, G. G., & York, W. (1996). *The webbook and the web forager: An information workspace for the world-wide web.* Paper presented at the Proceedings of CHI'96, ACM Conference on Human Factors in Computing Systems, New York.

Casner, S. (1991, April 2). Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics, 10*, 111–151.

Chi, E. H., & Card, S. K. (1999). *Sensemaking of evolving websites using visualization spreadsheets.* Paper presented at the Infovis 1999, IEEE Conference on Information Visualization 1999, San Francisco.

Chi, E. H., & Riedl, J. T. (1998). An operator interaction framework for visualization spreadsheets. Paper presented at the *Proceedings of InfoVis'98, IEEE Symposium on Information Visualization*.

Chuah, M. C., Roth, S. F., Mattis, J., & Kolojejchick, J. A. (1995). Sdm: Malleable information graphics. Paper presented at the *Proceedings of InfoVis'95, IEEE Symposium on Information Visualization*, New York.

Cleveland, W. S., & McGill, M. E. (1988). *Dynamic graphics for statistics*. Pacific Grove, California: Wadsworth and Brooks/Cole.

Cruz, I. F., & Tamassia, R. (1998). *Graph drawing tutorial*. Retrieved (date, from http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf.

Eick, S. G., Steffen, J. L., & Sumner, E. E. (1992, November). Seesoft—a tool for visualizing software. *IEEE Transactions on Software Engineering, 18*(11), 957–968.

Eick, S. G., & Wills, G. J. (1993, October 25–29). Navigating large networks with hierarchies. Paper presented at the *Proceedings of IEEE Visualization'93 Conference*, San Jose, CA.

Engelhardt, Y., Bruin, J. D., Janssen, T., & Scha, R. (1996). The visual grammar of information graphics. In S. University (Ed.), *Artificial intelligence in design workshop notes* (pp. 24–27).

Fairchild, K. M., Poltrock, S. E., & Furnas, G. W. (1988). Semnet: Three-dimensional representations of large knowledge bases. In R. Guindon (Ed.), *Cognitive science and its applications for human-computer interaction* (pp. 201–233). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Feiner, S., & Beshers, C. (1990). Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *ACM symposium on User Interface Software*.

Freeman, E., & Fertig, S. (1995). Lifestreams: Organizing your electronic life. Paper presented at the *Proceedings of AAAI Fall Symposium on AI Applications in Knowledge Navigation*.

Furnas, G. W. (1981). The fisheye view: A new look at structured files. In B. Shneiderman (Ed.), *Readings in information visualization: Using vision to think* (pp. 312–330). San Francisco: Morgan Kaufmann Publishers, Inc.

Healey, C. G., Booth, K. S., & Enns, J. T. (1995). High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction, 3*(2), 107–135.

Inselberg, A. (1997). Multidimensional detective. Paper presented at the *Proceedings of InfoVis'97, IEEE Symposium on Information Visualization, IEEE Information Visualization*.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates : A tool for visualizing multi-dimensional geometry. Paper presented at the *Proceedings of IEEE Visualization'90 Conference*, Los Alamitos, CA.

Keahey, T. A. (2001, October 22–23, 2001). *Getting along: Composition of visualization paradigms.* Paper presented at the Infovis 2001, IEEE Information Visualization 2001, San Diego, California.

Keim, D. A., & Kriegel, H.-P. (1994, September). Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 40–49.

Kosslyn, S. M. (1994). *Image and brain: The resolution of the imagery debate*. Cambridge, MA: The MIT PRess.

Lamping, J., & Rao, R. (1994). *Laying out and visualizing large trees using a hyperbolic space.* Paper presented at the Proceedings of UIST'94, ACM Symposium on User Interface Software and Technology.

Leung, Y. K., & Apperley, M. D. (1994, June). A review and taxonomy of distortion-orientation presentation techniques. *ACM Transactions on Computer-Human Interaction, 1*(2), 126–160.

Lin, X., Soergel, D., & Marchionini, G. (1991). *A self-organizing semantic map for information retrieval.* Paper presented at the Proceedings of SIGIR'91, ACM Conference on Research and Development in Information Retrieval, Chicago, IL.

MacEachren, A. M. (1995). *How maps work*. New York: The Guilford Press.

Mackinlay, J. D. (1986a). *Automatic design of graphical presentations.* Unpublished doctoral dissertation, Stanford University, California.

Mackinlay, J. D. (1986b). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics, 5*(2), 110–141.

Mackinlay, J. D., Robertson, G. G., & Card, S. K. (1991). *The perspective wall: Detail and context smoothly integrated.* Paper presented at the Proceedings of CHI'91, ACM Conference on Human Factors in Computing Systems, New York.

McCormick, B. H., & DeFanti, T. A. (1987, November 6). Visualization is scientific computing. *Computer Graphics, 21*.

Munzner, T., & Burchard, P. (1995, December 14–15). *Visualizing the structure of the world wide web in 3d hyperbolic space.* Paper presented at the Proceedings of VRML '95.

Playfair, W. (1786). *The commercial and political atlas*. London.

Resnikoff, H. L. (1989). *The illusion of reality*. New York: Springer-Verlag.

Risch, J. S., Rex, D. B., Dowson, S. T., Walters, T. B., May, R. A., & Moon, B. D. (1997). *The starlight information visualization system.* Paper presented at the Proceedings of IEEE International Conference on Information Visualization, London, England.

Robertson, G. G., Card, S. K., & Mackinlay, J. D. (1989). *The cognitive coprocessor for interactive user interfaces.* Paper presented at the Proceedings of UIST'89, ACM Symposium on User Interface Software and Technology.

Robertson, G. G., Card, S. K., & Mackinlay, J. D. (1993). Information visualization using 3d interactive animation. *Communications of the ACM, 36*(4), 57–71.

Robin, H. (1992). *The scientific image: From cave to computer*. New York: H. N. Abrams, Inc.

Roth, S. F., & Mattis, J. (1990). *Data characterization for intelligent graphics presentation.* Paper presented at the Proceedings of CHI'90, ACM Conference on Human Factors in Computing Systems, New York.

Russell, D. M., Stefik, M. J., Pirolli, P., & Card, S. K. (1993). *The cost structure of sensemaking.* Paper presented at the Proceedings of INTERCHI'93, ACM Conference on Human Factors in Computing Systems, Amsterdam.

Shneiderman, B. (1992). Tree visualization with tree-maps: A 2-dimensional space filling approach. *ACM Transactions on Graphics, 11*(1), 92–99.

Shneiderman, B., & Wattenberg, M. (2001). *Ordered tree layouts.* Paper presented at the IEEE Symposium on Information Visualization, San Diego, California.

Spence, R. (2000). *Information visualization*. Harlow, England: Addison-Wesley.

Spence, R., & Apperley, M. (1982). Data base navigation: An office environment for the professional. *Behavior and Information Technology, 1*(1), 43–54.

Tamassia, R. (1996, December 4). Strategic directions in computational geometry working group report. *ACM Computing Surveys, 28*, 591– 606.

Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1990). *Envisioning information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1997). *Visual explanations: Images and quantities, evidence and narrative*. Cheshire, CT: Graphics Press.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.

Tweedie, L. A., Spence, R., Dawkes, H., & Su, H. (1996). *Externalising abstract mathematical models.* Paper presented at the Proceedings of CHI'96, ACM Conference on Human Factors in Computing Systems.

Ware, C. (2000). *Information visualization: Perception for design*. San Francisco: Morgan Kaufmann Publishers.

·27·

GROUPWARE AND COMPUTER-SUPPORTED COOPERATIVE WORK

Gary M. Olson and Judith S. Olson
University of Michigan

Introduction **546**
Adopting Groupware in Context **546**
Technical infrastructure **547**
Communication Tools **547**
E-mail 547
Conferencing Tools: Voice, Video, Text 548
Blogs 550
Coordination Support **550**
Meeting Support 550
Workflow 551
Group Calendars 551
Awareness 551
Information Repositories **552**
Repositories of Shared Knowledge 552
Wikis 552
Capture and Replay 552
Social Computing **553**
Social Filtering, Recommender Systems 553
Trust of People via the Technology 553
Computer Supported Cooperative Learning **553**
Integrated Systems **553**
Media Spaces 553
Collaborative Virtual Environments 554
Collaboratories 554
Conclusions **554**
Acknowledgments **554**
References **554**

INTRODUCTION

Computing and communication technologies have provided us with useful and powerful information resources, remote instruments, and tools for interacting with each other. These possibilities have also led to numerous social and organizational effects. These tools are of course just the latest in a long line of modern technologies that have changed human experience. Television and radio long ago broadened our awareness of and interest in activities all over the world. The telegraph and telephone enabled new forms of organization to emerge. The new technologies of Computer-Supported Cooperative Work (CSCW)[1] are giving us greater geographical and temporal flexibility in carrying out our activities. They have also given us new modes of socializing.

Groupware is software designed to run over a network in support of the activities of a group or organization. These activities can occupy any of several combinations of same/different places and same/different times. Groupware has been designed for all four of these combinations. Early groupware applications tended to focus on only one of these cells, but more recently groupware that supports several cells and the transitions among them has emerged. We also do not think of groupware as only dealing with groups. Both the individual members of groups and the organizations in which they are embedded affect and are affected by groupware.

CSCW emerged as a formal field of study in the mid-1980s, with conferences, journals, books, and university courses appearing that used this name. There were a number of important antecedents. The earliest efforts to create groupware used time-shared systems but were closely linked to the development of key ideas that propelled the personal computer revolution. Vannever Bush (1945) described a vision of something similar to today's World Wide Web in an influential essay published shortly after the end of World War II. Doug Engelbart's famous demonstration at the 1968 IFIPS meeting in San Francisco included a number of key groupware components (see Engelbart & English, 1968). These components included support for real-time face-to-face meetings, audio and video conferencing, discussion databases, information repositories, and workflow support. Group decision-support systems and computer-supported meeting rooms were explored in a number of business schools (see McLeod, 1992; Kraemer & Pinsonneault, 1990). Work on office automation included many groupware elements, such as group workflow management, calendaring, e-mail, and document sharing (Ellis & Nutt, 1980). A good summary of early historical trends as well as reprints of key early articles appear in Grief's (1988) important anthology of readings.

Today there are a large number of commercial groupware products. In addition, groupware functions are increasingly appearing as options in operating systems or specific applications (e.g., access to meeting-support tools within Microsoft Office products). Groupware functionality has become widespread and familiar. However, there are still many research issues about how to design such systems and what effects they have on the individuals, groups, and organizations that use them.

ADOPTING GROUPWARE IN CONTEXT

Groupware systems are intended to support groups, which are usually embedded in an organization. As a result, there are a number of issues that bear on groupware success. In a justly famous set of papers, Grudin (1988, 1994) pointed out a number of problems that groupware systems have (see also Markus & Connolly, 1990). In brief, he pointed out that developers of groupware systems need to be concerned about the following issues (Grudin, 1994, p. 97):

1. Disparity in work and benefit. Groupware applications often require additional work from individuals who do not perceive a direct benefit from the use of the application.
2. Critical mass and Prisoner's dilemma problems. Groupware may not enlist the "critical mass" of users required to be useful, or can fail because it is never in any one individual's advantage to use it.
3. Disruption of social processes. Groupware can lead to activity that violates social taboos, threatens existing political structures, or otherwise de-motivates users crucial to its success.
4. Exception handling. Groupware may not accommodate the wide range of exception handling and improvisation that characterizes much group activity.
5. Unobtrusive accessibility. Features that support group processes are used relatively infrequently, requiring unobtrusive accessibility and integration with more heavily used features.
6. Difficulty of evaluation. The almost insurmountable obstacles to meaningful, generalizable analysis and evaluation of groupware prevent us from learning from experience.
7. Failure of intuition. Intuitions in product development environments are especially poor for multi-user applications, resulting in bad management decisions and error-prone design processes.
8. The adoption process. Groupware requires more careful implementation (introduction) in the workplace than product developers have confronted.

However, there are reasons for optimism. In a recent survey of the successful adoption of group calendaring in several organizations, Palen and Grudin (2002) observed that organizational conditions in the 1990s were much more favorable for the adoption of group tools than they were in the 1980s. Further, the tools themselves had improved in reliability, functionality, and usability. There is increased "collaboration readiness" and "collaboration technology readiness." But there are still significant challenges in supporting group work at a distance (Olson & Olson, 2000).

[1]This name contains several anachronisms. "Computer" is no longer confined to the familiar desktop device. "Cooperative" does not mean we ignore competitive uses of technology. "Work" is only one of a number of venues that CSCW researchers study.

TECHNICAL INFRASTRUCTURE

Groupware requires networks, and network infrastructure is a key enabler as well as a constraint on groupware. A number of advanced networks are exploring the issues of supporting high-end users (e.g., Abilene, abilene.internet2.edu; National LambdaRail, www.nlr.net; HOPI, networks.internet2.edu/hopi). Wireless networking technology gives users more flexibility. Good access to the Internet is now common in homes, hotels, coffee shops, airports, and many other places. Networking infrastructure is spreading throughout the world. However, heterogeneity in network conditions across both space and time still remains a major technical challenge. For instance, doing web conferencing when some participants are on slow dial-up lines and others are on fast advanced networks requires special coordination. Tanenbaum (2003) is a good resource on the latest developments in networking.

The World Wide Web and its associated tools and standards have had a major impact on the possibilities for groupware (Schatz & Hardin, 1994; Berners-Lee, 1999). Early groupware mostly consisted of stand-alone applications that had to be downloaded and run on each client machine. Increasingly, group tools are written for the web, requiring only a web browser and perhaps some plug-ins. This makes it much easier for the user, and also helps with matters such as version control. It also enables better interoperability across hardware and operating systems.

Security on the Internet is a major challenge for groupware. In some sense the design of Internet protocols are to blame, since the Internet grew up in a culture of openness and sharing (Longstaff et al., 1997; Abbate, 1999; Tanenbaum, 2003). E-commerce and sensitive application domains like medicine have been a driver for advances in security, but there is still much progress to be made (Longstaff et al., 1997; Camp, 2000). Coping with firewalls that block access to certain kinds of organizations can limit the flexibility of web conferencing.

Personal computing was a great enabler of collaborative applications. Today we are liberated from the desktop. Laptops, personal digital assistants, wearables, and cell phones provide access to information and people from almost anywhere. More and more applications are being written to operate across these diverse environments (e.g., Tang et al., 2001; Starner & Rhodes, 2004). These devices vary in computational power, display size and characteristics, network bandwidth, and connection reliability, providing interesting technical challenges to make them all interoperate smoothly. For instance, accessing websites from a cell phone requires special user interface methods to make the tiny displays usable.

Additional flexibility is being provided by the development of infrastructure that lies between the network itself and the applications that run on client workstations, called "middleware." This infrastructure makes it easier to link together diverse resources to accomplish collaborative goals. For instance, the emerging Grid technologies allow the marshalling of powerful, scattered computational resources (Foster & Kesselman, 2004). Middleware provides such services as identification, authentication, authorization, directories, and security in uniform ways that facilitate the interoperability of diverse applications. All of these technical elements are components of cyberinfrastructure (Atkins et al., 2003). There is considerable interest in the development of this infrastructure because of its large impact on research, education, and commerce.

COMMUNICATION TOOLS

We now turn to a review of specific kinds of groupware, highlighting their various properties and uses. We have grouped this review under several broad headings. We do not aim to be exhaustive, but rather seek to illustrate the variety of kinds of tools that have emerged to support human collaborative activities over networked systems. We also highlight various research issues pertaining to these tools.

E-mail

E-mail has become a ubiquitous communication tool. The early adoption of standards made it possible for messages to be exchanged across networks and different base machines and software applications. E-mail is now also done from cell phones, PDAs, television sets, and kiosks in public sites. Documents of many types can be easily exchanged. Because of its widespread use, it is often called the first successful groupware application (Sproull & Kiesler, 1991; Satzinger & Olfman, 1992; O'Hara-Devereaux & Johnson, 1994; Anderson, Bikson, Law, & Mitchell, 1995). Indeed, it has become so successful that e-mail overload has become a major problem (Whittaker & Sidner, 1996). And of course it has become a vector for viruses, worms, and other invasive software.

Researchers have shown that this widespread use has had a number of effects on how people behave. It has had large effects on communication in organizations: it changes the social network of who talks to whom (Sproull & Kiesler, 1991; DeSanctis, Jackson, Poole, & Dickson, 1996), the power of people who formerly had little voice in decisions (Finholt, Sproull, & Kiesler, 1990), and the tone of what is said and how it is interpreted (Sproull & Kiesler, 1991). For example, with e-mail, people who were shy found a voice; they could overcome their reluctance to speak to other people by composing text, not speech to another face. This invisibility, however, also has a more general effect—without the social cues in the recipient's face being visible to the sender, people will "flame," send harsh or extremely emotive (usually negative) messages (Arrow et al., 1996; Hollingshead, McGrath, & O'Connor, 1993).

As with a number of other "designed" technologies, people use e-mail for things other than the original intent. People use it for managing time, reminding them of things to do, and keeping track of steps in a workflow (Mackay, 1989; Carley & Wendt, 1991; Whittaker & Sidner, 1996). But because e-mail was not designed to support these tasks, it does not do it very well; people struggle with reading signals about whether they have replied or not (and to whom it was cc'd); they manage folders poorly for reminding them to do things, and so forth.

In addition, because e-mail is so widespread, and it is easy and free to distribute a single message to many people, people

experience information overload. Many people get hundreds of e-mail messages each day, many of them mere broadcasts of things for sale or events about to happen, much like "classifieds" in the newspaper. Several early efforts to use artificial intelligence techniques to block and/or sort incoming e-mail were tried, and this has continued to be a very active area of work (Malone, Grant, Lai, Rao, & Rosenblitt, 1987; Winograd, 1988). There are two broad classes of uses of e-mail filters. One use is to automatically sort incoming mail into useful categories. This is relatively easy for mail that has simple properties, such as a person's name. It is more difficult for subtle properties. The other major use is to weed out unwanted mail, such as spam. The state-of-the-art in spam filtering is in the range of 80–90% effectiveness in 2005 (e.g., Federal Trade Commission, 2005). Such filters are so good that many institutions automatically filter mail as it comes in to the organization's gateway, sparing users the need to do it in their own clients. Similarly, many clients now come with built-in spam filters that can be tuned by the user (e.g., Google's Gmail).

These problems have led to the "reinvention" of e-mail (Whittaker, Bellotti, & Moody, 2005). For example, given that e-mail is often used in the context of managing projects, systems have been explored that have a more explicit scheme for task management (Whittaker, 2005; Bellotti, Ducheneaut, Howard, Smith, & Grinter, 2005). To deal with problems of e-mail overload, new schemes for filtering e-mail have been explored, such as routing messages differently to different kinds of clients (e.g., cell phone vs. desktop machine; see Schmandt & Marti, 2005). Another approach has been to explore pricing mechanisms for e-mail that are analogous to pricing for regular mail (Kraut, Sunder, Telang, & Morris, 2005). In such schemes one would pay to send e-mail, with higher prices presumably indicating higher priority, analogous to the difference between first class postage and bulk rates. These schemes are currently exploratory, but are likely to result in new options in future e-mail clients.

Kraut et al. (1998) reported that greater Internet use, which in their sample was mostly e-mail, led to declines in social interactions with family members and an increase in depression and loneliness. Not surprising, these results triggered widespread discussion and debate, both over the substance of the results and the methods used to obtain them. Kraut, Gergle, and Fussell (2002) reported new results that suggest these initial negative effects may not persist. Interpersonal communication is one of the principal uses of the Internet, and the possible implications of this kind of communication for social life are important to understand. Indeed, Putnam (2000) has wondered whether the Internet can be a source of social cohesiveness. These kinds of questions need to be addressed by additional large-scale studies of the kind carried out by Kraut and his colleagues (see Resnick, 2002).

Conferencing Tools: Voice, Video, Text

There are many options available today for on-line conferencing among geographically dispersed members of a group. So-called computer-mediated communication (CMC) has become widespread. There are three principal modes of interaction, but each has numerous subtypes:

Video + Audio

Full-scale video conferencing room; many options for specific design

Individual desktop video; many options for quality, interface

Audio

Phone conference

Voice over IP (see Federal Communications Commission, 2005)

Text

Instant messaging, chat, SMS on mobile phones

There are many studies that compare face-to-face with various forms of CMC. There are some clear generalizations from such work. The main one is that CMC is more difficult to do than FTF, and requires more preparation and care (Hollingshead et al., 1993; McLeod, 1992; Olson, Olson, & Meader, 1995; Siegel, Dubrovsky, Kiesler, & McGuire, 1986; Straus, 1996, 1997; Straus & McGrath, 1994). A variety of things that come for free in FTF are either difficult to support or outright missing in CMC (Kiesler & Cummings, 2002). Backchannel communication, which is important for modulating conversation, is either weak or nonexistent in CMC. Paralinguistic cues that can soften communication are often missing. Participants in CMC tend to have an informational focus, which means there is usually less socializing, less small talk. Over time this can lead to poorer social integration and organizational effectiveness (Nohria & Eccles, 1992).

CMC often introduces delay. This is well-known to be very disruptive to communication (Egido, 1988; Krauss & Bricker, 1966; O'Conaill, Whittaker, & Wilbur, 1993; Ruhleder & Jordan, 2001; Tang & Isaacs, 1993). Participants will communicate less information, be more frustrated with the communication, and actually terminate communication sessions sooner. Delay can be managed, but it takes special care among the participants and turn-taking widgets in the interface of the tools being used. For instance, if there is delay, then full-duplex open communication will not work, since participants will step all over each other's communication. Either the participants must use a social protocol (e.g., like that used in radio communications with spacecraft), or they must employ a mike-passing procedure with interface indications of who wants to talk next.

While it might seem desirable to always have the maximum communication and tool support possible, it is not always possible or even necessary to do so. Research shows that effective real-time collaboration can take place under a number of different arrangements, depending on the task, the characteristics of the participants, the specific geographical dispersion of the participants, and the processes employed to manage the interactions. There are also organizational effects, especially when the real-time collaborations are embedded in ongoing activities, as they almost always are.

For instance, early work (Williams, 1977) showed that, in referential communication tasks, full-duplex audio is just as ef-

fective as FTF. Subsequent research comparing audio and video conferencing (see Finn, Sellen, & Wilbur, 1997; Cadiz et al., 2000 recently found similar results for a tutored video-instruction task) showed that for many tasks audio is sufficient, that video adds nothing to task effectiveness though participants usually report they are more satisfied with video. There are important exceptions, however. Negotiation tasks are more effective with video (Short, Williams, & Christie, 1976). This is probably because the more subtle visual cues to the participants' intentions are important in this kind of task. Further, Veinott, Olson, Olson, and Fu (1999) found that when participants have less common ground video helps. In their case, participants were non-native speakers of English who were doing the task in English. For native speakers, video was no better than audio, but non-native speakers did better when they had video. Again, visual cues to comprehension and meaning likely played an important role. Recently, an experimental study by Daly-Jones, Monk, and Watts (1998) showed that high-quality video resulted in greater conversational fluency over just high-quality audio, especially as group size increased. There was also a higher rated sense of presence in the video conditions.

An important lesson to draw from this literature is that there are two broad classes by which we might assess whether video is important in real-time collaboration. On the one hand, except for tasks like negotiation or achieving common ground, groups are able to get their work done effectively with just high-quality audio. However, for things like satisfaction, conversational fluency, and a sense of presence, video adds value. These kinds of factors might be very important for long-term organizational consequences like employee satisfaction. As of yet, no long-term studies have been done to examine this conjecture.

Audio quality is critical. Ever since early literature review (Egido, 1988), it's been reported over and over again that if the audio is of poor quality participants will develop a workaround. For instance, if the audio in a video conferencing system or in a web conferencing system is poor quality, participants will turn to a phone conference.

The social ergonomics of audio and video are also keys to their success. Many of the failures of audio conferencing, especially over the Internet, result from poor-quality microphones, poor microphone placement, poor speakers, and interfering noises like air conditioning. Getting these details right is essential. Similarly, for video, camera placement can matter a lot. For instance, Huang, Olson, and Olson (2002) found that a camera angle that makes a person seem tall (as opposed to actually being tall) affects how influential a person is in a negotiation task. Apparent height matters a lot. Other aspects of camera placement or arrangement of video displays make a big difference as well, but are not well known.

An exception is eye contact, where studies of FTF communication show that eye contact is a key linguistic and social mediator of communication (Argyle & Cook, 1976; Kendon, 1967). It is very difficult to achieve eye contact in CMC systems. Many attempts have been made (Gale & Monk, 2000; Grayson & Monk, 2003; Monk & Gale, 2002; Okada, Maeda, Ichicawaa, & Matsushita, 1994; Vertegaal, 1999; Vertegaal, Slagter, van der Veer, & Nijholt, 2001), and at least the subjective reports are that these can be effective. But these all require special equipment or setups. And they don't scale very well to multiparty sessions.

While for most situations having at least high-quality audio is essential, there are some special cases where a text-based channel, like chat or instant messaging, can work fine. For instance, in the Upper Atmospheric Research Collaboratory (UARC, later known as the "Space Physics and Aeronomy Research Collaboratory," or SPARC), a chat system worked very well for carrying out geographically distributed observational campaigns, since the flow of events in these campaigns were relatively slow (campaigns went on for several days, key events would take many minutes to unfold). McDaniel, Olson, and Magee (1996) compared chat logs with earlier FTF conversations at a remote site and found many elements of them very similar, including informal socializing. But this kind of ongoing scientific campaign is very unlike the interactions that take place in a typical meeting.

Instant messaging is a new communication modality that is making substantial inroads into organizations. Muller, Raven, Kogan, Millen, and Carey (2003) found in a survey study of three organizations that the introduction of instant messaging led to significantly less use of such communication channels as e-mail, voice-mail, telephone, teleconference, pager, and face-to-face. They also found that instant messaging was used for "substantive business purposes." Furthermore, in one of the organizations where they surveyed users after 24 months of usage they found that the substantive reasons for using IM increased. In a study of IM logs in an organization, Isaacs, Walendowski, Whittaker, Schiano, and Kamm (2002) found that a large proportion of IM conversations involved "complex work discussions." They found that IM users seldom switched to another communication channel once they were engaged in IM. Nardi, Whittaker, and Bradner (2000) observed in a field study that workers used IM for a variety of purposes, not just for information exchange. Such matters as quick questions, scheduling, organizing social interactions, and keeping in touch with others were common uses of IM. Thus, IM has emerged as a significant communication medium in the workplace, and is used even when other, richer communication channels were available.

While IM is a relatively new phenomenon in the workplace, it is clearly established as a useful and widely used tool outside the workplace. This will undoubtedly assist in the development of more sophisticated versions of the tool, as well as its integration into on-line conferencing systems. There is clearly much promise here. We have noticed, for example, that during online conferences IM or chat serves as a backchannel for side conversations or debugging, an extremely useful adjunct to the core audio or video communication taking place in such conferences.

The other key feature of successful remote meetings is the ability to share the objects they are talking about, such as the agenda, the to-do list, the latest draft of a proposal, a view of an object to be repaired, and so on. Many researchers (Fussell, Kraut, & Siegel, 2000; Karsenty, 1999; Kraut, Fussell, & Siegel, 2003; Kraut et al., 2002; Luff, Heath, Kuzuoka, Hindmarsh, & Oyama, 2003; Nardi et al., 1993; Whittaker, 2003; Whittaker, Geelhoed, & Robinson, 1993) have provided experimental evidence of the value of a shared workspace for synchronous audio-supported collaboration. More traditional video conferencing

technologies often offer an "object camera," onto which the participants can put a paper agenda, Powerpoint slides, or a manufactured part. More generally, any form of video can also be used to share work objects (Fussell et al., 2000; Nardi et al., 1993). For digital objects, there are now a number of products that will allow meeting participants to share the screen or, in some cases, the remote operation of an application. Some companies are using electronic whiteboards, both in a collocated meeting and in remote meetings to mimic the choreography of people using a physical whiteboard. In some "collaboratories," scientists can even operate remote physical instruments from a distance and jointly discuss the results.

Blogs

Weblogs, or more commonly called "blogs," have burst upon the Internet scene in recent years. Blogging software that makes it easy to put up multimedia content has led people to set up sites for all manner of purposes. A site can contain text, pictures, movies, and audio clips. A common social purpose is to keep an on-line diary. Another is to provide commentary on a topic of interest. For instance, blogs played a major role in the 2004 election (Adamic & Glance, 2005). Nardi, Schiano, Gumbrecht, and Swartz (2004) studied why people blog, as it's sometimes puzzling that people would essentially share personal or private information about themselves through the web.

COORDINATION SUPPORT

Meeting Support

An early and popular topic in CSCW was the support of face-to-face meetings. A number of systems were developed and tested. While of late the focus has shifted to the support of geographically distributed meetings, the early work on meeting support led to some important and useful conclusions.

Some meeting-support software imposed structure on the process of the meeting, embodying various brainstorming and voting procedures. Group Decision Support Systems (GDSSs) arose from a number of business schools, focusing on large meetings of stakeholders intent on going through a set series of decisions, such as prioritizing projects for future funding (Nunamaker, Dennis, Valacich, Vogel, & George, 1991). With the help of a facilitator and some technical support, the group was led through a series of stages: brainstorming without evaluating, evaluating alternatives from a variety of positions, prioritizing alternatives, and so on. These meetings were held in specialized rooms in which individual computers were embedded in the tables, networked to central services, and summary displays shown "center stage." A typical scenario involved individuals silently entering ideas into a central repository, and after a certain amount of time, they were shown ideas one at a time from others and asked to respond with a new idea triggered by that one. Later, these same ideas were presented to the individuals who were then asked to rank or rate them according to some fixed criterion, like cost. Aggregates of individuals' opinions were computed, discussed further and presented for vote. The system applied computational power (for voting and rating mechanisms), and networking control (for parallel input) to support typically weak aspects of meetings. These systems were intended to gather more ideas from participants, since one didn't have to wait for another to stop speaking in order to get a turn. And, anonymous voting and rating was intended to insure equal participation, not dominated by those in power.

Evaluations of these GDSSs have been reviewed producing some generalizations about their value (McLeod, 1992; Kraemer & Pinsoneault, 1990; Hollingshead et al., 1993). The systems indeed fulfill their intentions of producing more ideas in brainstorming and having more evaluative comments because of anonymity. Decisions are rated as higher in quality, but the meetings take longer and the participants are less satisfied than those in traditional meetings.

A second class of technologies to support real-time meetings is less structured, more similar to individual workstation support. In these systems, groups are allowed access to a single document or drawing, and can enter and edit into them simultaneously at will. Different systems enforce different "locking" mechanisms (e.g., paragraph or selection locking) so that one person does not enter while another deletes the same thing (Ellis, Gibbs, & Rein, 1991). Some also allow parallel individual work, where participants view and edit different parts of the same document, but can also view and discuss the same part as well. This kind of unstructured shared editor has been shown to be very effective for certain kinds of free-flowing meetings, like design or requirements meetings (Olson, Olson, Storrosten, & Carter, 1993). The rated quality of the meeting products (e.g., a requirements document or plan) was higher when using these technologies than with traditional whiteboard or paper-and-pencil support, but like working in GDSSs, people were slightly less satisfied. The lower satisfaction here and with GDSSs may reflect the newness of the technologies; people may not have yet learned how to persuade, negotiate, or influence each other in comfortable ways, to harness the powers inherent in the new technologies.

These new technologies did indeed change the way in which people worked. They talked less and wrote more, building on each other's ideas instead of generating far-reaching other ideas. The tool seemed to focus the groups on the core ideas, and keep them from going off on tangents. Many participants reported really liking *doing* work *in* the meetings rather than spending time only *talking about* the work.

A third class of meeting room support appears in electronic whiteboards. For example, the LiveBoard (Elrod et al., 1992), SoftBoard and SmartBoard are approximately $4' \times 6'$ rear-projection surfaces that allow pen input, much the way a whiteboard or flipchart does. People at Xerox PARC and Boeing have evaluated the use of these boards in meetings in extended case studies. In both cases, the board was highly valued because of its computational power and the fact that all could see the changes as they were made. At both sites, successful use required a facilitator who was familiar with the applications running to support the meeting. At Xerox, suggestions made in the meeting about additional functionality were built into the system so

that it eventually was finely tuned support for their particular needs (Moran et al., 1996). For example, they did a lot of list making of freehand text items. Eventually, the board software recognized the nature of a list and an outline, with simple gestures changing things sensibly. For example, if a freehand text item was moved higher in a list, the other items adjusted their positions to make room for it. The end product was not only a set of useful meeting tools, but also a toolkit to allow people to build new meeting widgets to support their particular tasks.

Meetings are important, though often despised, organizational activities. Laboratory research of the kind just reviewed has shown quite clearly that well-designed tools can improve both work outcomes and participant satisfaction. However, meetings in organizations seldom use such tools. Inexpensive mobile computing and projection equipment combined with many commercial products mean that such tools are within reach of most organizations. But not having these elements readily available in an integrated way probably inhibits their widespread adoption.

While traditional meetings are often viewed as wasteful and frustrating, there can be huge benefits to working together in collocated environments. Kiesler and Cummings (2002) reviewed a number of the characteristics of physical collocation that can benefit performance. In a detailed study of one such situation, Teasley, Covi, Krishnan, and Olson (2002) found that "radical collocation," in which software development teams worked together in a dedicated project room for many weeks, dramatically improved their productivity. Reasons for this included the constant awareness of each other's work status, the associated ability to instantly work on an impasse as a group, and the availability of rich shared artifacts generated by the project.

Workflow

Workflow systems lend technology support to coordinated asynchronous (usually sequential) steps of activities among team members working on a particular task. For example, a workflow system might route a travel reimbursement voucher from the traveler to the approving party to the accounts payable to the bank. The electronic form would be edited and sent to the various parties, their individual to-do-lists updated as they received and/or completed the tasks, and permissions and approval granted automatically as appropriate (e.g., allowing small charges to an account if the charges had been budgeted previously or simply if there was enough money in the account). Not only is the transaction flow supported, but also records are often kept about who did what and when they did it. It is this later feature that has potentially large consequences for the people involved, discussed later.

These workflow systems were often the result of work reengineering efforts, focusing on making the task take less time and to eliminate the work that could be automated. Not only do workflow systems therefore have a bad reputation in that they often are part of workforce reduction plans, but also for those left, their work is able to be monitored much more closely. The systems are often very rigid, requiring, for example, all of a form to be filled in before it can be handed off to the next in the chain. They often require a great deal of rework because of this inflexibility. It is because of the inflexibility and the potential monitoring that the systems fall into disuse (e.g., Abbott & Sarin, 1994).

The fact that workflow can be monitored is a major source of user resistance. In Europe, such monitoring is illegal, and powerful groups of organized workers have made sure that such capabilities are not in workflow systems (Prinz & Kovenbach, 1996). In the United States, it is not illegal, but many employees complain about its inappropriate use. For example, in one software engineering team where workflow had just been introduced to track bug reports and fixes, people in the chain were sloppy about noting who they had handed a piece of work off to. When it was discovered that the manager had been monitoring the timing of the handoffs to assign praise or blame, the team members were justifiably upset (Olson & Teasley, 1996). In general, managerial monitoring is a feature that is not well received by people being monitored (Markus, 1983). If such monitoring is mandated, workers' behavior will conform to the specifics of what is being monitored (e.g., time to pass an item off to the next in the chain) rather than perhaps to what the real goal is (e.g., quality as well as timely completion of the whole process).

Group Calendars

A number of organizations have now adopted online calendars, mainly in order to view people's schedules to arrange meetings. The calendars also allow a form of awareness, allowing people to see if a person who is not present is expected back soon. Individuals benefit only insofar as they offload scheduling meetings to others, like to an administrative assistant, who can write as well as read the calendar. And, in some systems the individual can schedule private time, blocking the time but not revealing to others his or her whereabouts. By this description, on-line calendaring is a classic case of what Grudin (1988) warned against, a misalignment of costs and benefits; the individual puts in the effort to record his/her appointments so that another, in this case a manager or coworker, can benefit from ease of scheduling. However, since the early introduction of electronic calendaring systems, many organizations have found successful adoption (Mosier & Tammaro, 1997; Grudin & Palen, 1995; Palen & Grudin, 2002). Apparently such success requires a culture of sharing and accessibility, something that exists in some organizations and not others (Lange, 1992; Ehrlich, 1987).

Awareness

In normal work, there are numerous occasions in which people find out casually whether others are in and, in some cases, what they are doing. A simple walk down the hall to a printer offers numerous glances into people's offices, noting where their coats are, whether others are talking, whether there is intense work at a computer, and so on. This kind of awareness is unavailable to workers who are remote. Some researchers have offered various technology solutions; some have allowed one to visually walk down the hall at the remote location, taking a five-second glance into each passing office (Bellotti & Dourish,

1997; Fish, Kraut, Root, & Rice, 1993). Another similar system, called "Portholes," provides periodic snapshots instead of full-motion video (Dourish & Bly, 1992). Because of privacy implications, these systems have had mixed success. The places in which this succeeds are those in which the individuals seem to have a reciprocal need to be aware of each other's presence, and a sense of cooperation and coordination. A contrasting case is the instant messaging (IM) system in which the user has control as to what state they wish to advertise to their partners about their availability. The video systems are much more lightweight to the user but more intrusive; the IM ones give the user more control but require intention in action. Another approach investigated by Ackerman, Starr, Hindus, and Mainwaring (1997) looked at shared audio as an awareness tool, though this too has privacy implications.

As mentioned earlier, instant messaging systems provide an awareness capability. Most systems display a list of "buddies" and whether they are currently on-line or not. Nardi et al. (2000) found that people liked this aspect of IM (see also Muller et al., 2003; Isaacs et al., 2002). And, since wireless has allowed constant connectivity of mobile devices like PDAs, this use of tracking others is likely to grow. But again, there are issues of monitoring for useful or insidious purposes, and the issues of trust and privacy loom large (see Godefroid, Herbsleb, Jagadeesan, & Li, 2000).

Another approach to signaling what one is doing occurs at the more micro level. And again, one captures what is easy to capture. When people are closely aligned in their work, there are applications that allow each to see exactly where in the shared document the other is working and what they are doing (Gutwin & Greenberg, 1999). If one is working nearby the other, this signals perhaps a need to converse about the directions each is taking. Empirical evaluations have shown that such workspace awareness can facilitate task performance (Gutwin & Greenberg, 1999).

Studies of attempts to carry out difficult intellectual work within geographically distributed organizations show that one of the larger costs of geographical distribution is the lack of awareness of what others are doing or whether they are even around (Herbsleb, Mockus, Finholt, & Grinter, 2000). Thus, useful and usable awareness tools that mesh well with trust and privacy concerns could be of enormous organizational importance. This is a rich research area for CSCW.

INFORMATION REPOSITORIES

Repositories of Shared Knowledge

In addition to sharing information generally on the web, in both public and intranet settings, there are applications that are explicitly built for knowledge sharing. The goal in most systems is to capture knowledge that can be reused by others, like instruction manuals, office procedures, training, and "boilerplates," or templates of commonly constructed genres, like proposals or bids. Experience shows, however, that these systems are not easy wins. Again, similar to the case of the on-line calendaring systems described above, the person entering information into the system is not necessarily the one benefiting from it. In a large consulting firm, where consultants were quite competitive in their bid for advancement, there was indeed negative incentive for giving away one's best secrets and insights (Orlikowski & Gash, 1994).

Sometimes subtle design features are at work in the incentive structure. In another adoption of Lotus Notes, in this case to track open issues in software engineering, the engineers slowly lost interest in the system because they assumed that their manager was not paying attention to their contributions and use of the system. The system design, unfortunately, made the manager's actual use invisible to the team. Had they known that he was reading daily what they wrote (though he never wrote anything himself), they would likely have continued to use the system (Olson & Teasley, 1996). A simple design change that would make the manager's reading activity visible to the team would likely have significantly altered their adoption.

The web of course provides marvelous infrastructure for the creation and sharing of information repositories. A variety of tools are appearing to support this. Of particular interest are open source tools that allow for a wider, more flexible infrastructure for supporting information sharing (see www.sakai.org). A major type of collaboratory (see below) are those that provide shared data repositories for a community of scientists. Systematic research on the use of such tools is needed.

Wikis

A wiki is a shared web space that can be edited by anyone who has access to it. They were first introduced by Ward Cunningham in 1995, but have recently become very popular. These can be used in a variety of ways, both for work and for fun. The most famous wiki is Wikipedia (www.wikipedia.org), an online encyclopedia where anyone can generate and edit content. It has grown to have millions of entries, and has versions in at least ten languages. A recent study carried out by *Nature* found that for science articles Wikipedia and the Encyclopedia Britannica were about equally accurate (Giles, 2005). Bryant, Forte, and Bruckman (2005) studied the contributors to Wikipedia, and suggested that a new publishing paradigm was emerging. Viegas, Wattenberg, and Dave (2004) developed imaginative visualizations of Wikipedia authoring and editing behavior over time.

Capture and Replay

Tools that support collaborative activity can create traces of that activity that later can be replayed and reflected upon. The Upper Atmospheric Research Collaboratory (UARC) explored the replay of earlier scientific campaign sessions (Olson et al., 2001), so that scientists could reflect upon their reactions to real-time observations of earlier phenomena. Using a VCR metaphor, they could pause where needed, and fast forward past uninteresting parts. This reflective activity could also engage new players who had not been part of the original session. Abowd (1999) has explored such capture phenomena in an educational experiment called Classroom 2000. Initial experiments focused on reusing educational sessions during the term in college courses. We do net yet fully understand the impact of such promising ideas.

SOCIAL COMPUTING

Social Filtering, Recommender Systems

We often find the information we want by contacting others. Social networks embody rich repositories of useful information on a variety of topics. A number of investigators have looked at whether the process of finding information through others can be automated. The kinds of recommender systems that we find on websites like Amazon.com are examples of the result of such research. The basic principle of such systems is that an individual will tend to like or prefer the kinds of things (e.g., movies, books) that someone who is similar to him/her likes. They find similar people by matching their previous choices. Such systems use a variety of algorithms to match preferences with those of others, and then recommend new items. Resnick and Varian (1997) edited a special issue of the Communication of the ACM on recommender systems that included a representative set of examples. Herlocker, Konstan, and Riedl (2001) used empirical methods to explicate the factors that led users to accept the advice of recommender systems. In short, providing access to explanations for why items were recommended seems to be the key. Cosley, Frankowski, Kiesler, Terveen, and Riedl (2005) studied factors that influence people to contribute data to recommender systems. Recommender systems are emerging as a key element of e-commerce (Schafer, Konstan, & Riedl, 2001). Accepting the output of recommender systems is an example of how people come to trust technical systems. This is a complex topic, and relates to issues like security that we briefly described earlier.

Trust of People via the Technology

It has been said that "trust needs touch," and indeed in survey studies, coworkers report that they trust those who are collocated more than those who are remote (Rocco, Finholt, Hofer, & Herbsleb, 2000). Interestingly, those who spend the most time on the phone chatting about non-work related topics with their remote coworkers show higher trust than those they communicate with using only fax and e-mail. But lab studies show that telephone interaction is not as good as face-to-face. People using just the telephone behave in more self-serving, less-trusting ways than they do when they meet face to face (Drolet & Morris, 2000).

What can be done to counteract the mistrust that comes from the impoverished media? Rocco (1998) had people meet and do a team-building exercise the day before they engaged in the social dilemma game with only e-mail to communicate with. These people, happily, showed as much cooperation and trust as those who discussed things face to face during the game. This is important. It suggests that if remote teams can do some face-to-face teambuilding before launching on their project, they will act in a trusting/trustworthy manner.

Since it's not always possible to have everyone on a project meet face to face before they launch into the work, what else will work? Researchers have tried some options, but with mixed success. Zheng, Bos, Olson, Gergle, and Olson (2001) found that using chat for socializing and sharing pictures of each other also led to trustful relations. Merely sharing a resume did not. When the text is translated into voice, it has no effect on trust, and when it is translated into voice and presented in a moving human-like face, it is even worse than text-chat. (Jensen, Farnham, Drucker, & Kollock, 2000; Kiesler, Sproull, & Waters, 1996). However, Bos, Gergle, Olson, and Olson (2001) found that interactions over video and audio led to trust, albeit of a seemingly more fragile form.

If we can find a way to establish trust without expensive travel, we are likely to see important productivity gains. Clearly the story is not over. However, we must not be too optimistic. In other tasks, video does not produce "being there." There is an overhead to the conversation through video; it requires more effort than working face to face (Olson et al., 1995). And, today's videos over the Internet are both delayed and choppy, producing cues that people often associate with lying. One doesn't trust someone who appears to be lying. Trust is a delicate emotion; today's video might not just do it in a robust enough fashion.

COMPUTER SUPPORTED COOPERATIVE LEARNING

Obviously the range of CSCW systems that we've been describing can be used for a variety of purposes. One special area that has emerged as a subfield with a distinctive identity is education, using the name Computer Supported Cooperative Learning (CSCL). This field first emerged in Europe, but has become quite widespread. This area has its own journals and national meetings (see Koschmann, 1996; Koschmann, Hall, & Miyake, 2002; Koschmann, Suthers, & Chan, 2005).

The emergence of this field has coincided with the emergence of a general trend in education toward collaborative learning, of using working with peers in groups as an effective tool for education (Slavin, 1994). The core idea is that by working together with peers the interactions over problem solving and other learning activities will occur at a level that is most easily understood and engaged by the learners. What CSCL adds to this is a focus on software that can facilitate collaboration, particularly across distance.

INTEGRATED SYSTEMS

Media Spaces

As an extension of video conferencing and awareness systems, some people have experimented with open, continuous audio and video connections between remote locations. In a number of cases, these experiments have been called "Media Spaces." For example, at Xerox, two labs were linked with an open video link between two commons areas (Olson & Bly, 1991), the two locations being Palo Alto, California, and Portland, Oregon. Evaluation of these experiments showed that maintaining organizational cohesiveness at a distance was much more difficult than when members are collocated (Finn et al., 1997). However, some connectedness was maintained. Where many of these early systems were plagued with technical difficulties, human factors limitations,

or very large communication costs, in today's situation it might actually be possible to overcome these difficulties, making media a possibility for connecting global organizations. A new round of experimental deployments with new tools is needed.

Collaborative Virtual Environments

Collaborative virtual environments are 3D embodiments of multi-user domains (MUDs). The space in which people interact is an analog of physical space, with dimensions, directions, rooms, and objects of various kinds. People are represented as avatars—simplified, geometric, digital representations of people, who move about in the 3D space (Singhal & Zyda, 1999). Similar to MUDs, the users in a meeting situation might interact over some object that is digitally represented, like a mock up of a real thing (e.g., an automobile engine, an airplane hinge, a piece of industrial equipment) or with visualizations of abstract data (e.g., a 3D visualization of atmospheric data). In these spaces, one can have a sense as to where others are and what they are doing, similar to the simplified awareness systems described above. In use, it is difficult to establish mutual awareness or orientation in such spaces (Hindmarsh, Fraser, Heath, Benford, & Greenhalgh, 1998; Park, Kapoor, & Leigh, 2000; Yang & Olson, 2002). There have even been some attempts to merge collaborative virtual environments with real ones, though with limited success so far (Benford, Greenhalgh, Reynard, Brown, & Koleva, 1998).

What people seem to want is more like the Holodek in "Star Trek." These environments are complicated technically, and perhaps even more complicated socially. In real life, we have developed interesting schemes that trigger behavior and interpretation of others' behavior as a function of real distance, a field called "Proxemics" (Hall, 1982). Only when these subtle behaviors are incorporated into the virtual environment will we have a chance of simulating appropriate interhuman behavior in the virtual 3D world.

Collaboratories

A collaboratory is a laboratory without walls (Finholt & Olson, 1997). From a National Research Council report, a collaboratory is supposed to allow "the nation's researchers [to] perform their research without regard to geographical location—interacting with colleagues, accessing instrumentation, sharing data and computational resources [and] accessing information in digital libraries" (National Research Council, 1993, p. 7). Starting in the early 1990s, these capabilities have been configured into support packages for a number of specific sciences (see Finholt, 2002). The Science of Collaboratories project (www.scienceofcollaboratories.org) has identified more than 200 existing collaboratories and is drawing lessons about why some succeed and others don't (Olson et al., 2004).

A number of companies have also experimented with similar concepts, calling them "virtual collocation." The goal there is to support geographically dispersed teams as they carry out product design, software engineering, financial reporting, and almost any business function. In these cases, suites of off-the-shelf groupware tools have been particularly important and have been used to support round-the-clock software development among overlapping teams of engineers in time zones around the world. (Carmel, 1999). There have been a number of such efforts, and it is still unclear as to their success or what features make their success more likely. (Olson & Olson, 2000).

CONCLUSIONS

Groupware functionality is steadily becoming more routine in commercial applications. Similarly, suites of groupware functions are being written into operating systems. Many of the functions we have described in this article are becoming ordinary elements of infrastructure in networked computing systems.

Prognosticators looking at the emergence of groupware and the convergence of computing and communication media have forecast that distance will diminish as a factor in human interactions (e.g., Cairncross, 1997). However, to paraphrase Mark Twain, the reports of distance's death are greatly exaggerated. Even with all our emerging information and communications technologies, distance and its associated attributes of culture, time zones, geography, and language will continue to affect how humans interact with each other. Emerging distance technologies will allow greater flexibility for those whose work must be done at a distance, but we believe (see Olson & Olson, 2000) that distance will continue to be a factor in understanding these work relationships.

ACKNOWLEDGMENTS

Preparation of this chapter was facilitated by several grants from the National Science Foundation (research grants IIS-9320543, IIS-9977923, ATM-9873025, IIS-0085951 and cooperative agreement IRI-9216848). We are also grateful to several anonymous reviewers for helpful comments on an earlier draft.

References

Abbate, J. (1999). *Inventing the Internet*. Cambridge, MA: MIT Press.

Abbott, K. R., & Sarin, S. K. (1994). Experiences with workflow management: Issues for the next generation. *Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 113–120). Chapel Hill, NC: ACM Press.

Abowd, G. D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal, 38*, 508–530.

Ackerman, M. S., Starr, B., Hindus, D., & Mainwaring, S. D. (1997). Hanging on the 'wire': A field study of an audio-only media

space. *ACM Transactions on Computer-Human Interaction, 4*(1), 39–66.

Adamic, L. A., & Glance, N. (2005). The political blogosphere and the 2004 U.S. election: Divided they blog. Presented at LinkKDD-2005, Chicago, IL. Retrieved March 13, 2007, from www.blogpulse.com/papers/2005/.

Anderson, R. H., Bikson, T.K., Law, S. A., & Mitchell, B. M. (1995). *Universal access to e-mail: Feasibility and societal implications*. Santa Monica, CA: Rand.

Argyle, M., & Cook, M. (1976). *Gaze and mutual gaze*. New York: Cambridge University Press.

Arrow, H., Berdahl, J. L., Bouas, K. S., Craig, K. M., Cummings, A, Lebei, L., McGrath, J. E., O'Connor, K. M., Rhoades, J. A., & Schlosser, A. (1996). Time, technology, and groups: An integration. *Computer Supported Cooperative Work, 4*, 253–261.

Atkins, D. E., Droegemeier, K. K., Feldman, S. I., Garcia-Molina, H., Klein, M. L., Messerschmitt, D. G., Messina, P., Ostriker, J. P., & Wright, M. H. (2003). *Revolutioning science and engineering through cyberinfrastructure*. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure. National Science Foundation. Arlington, VA.

Bellotti, V., & Dourish, P. (1997). Rant and RAVE: Experimental and experiential accounts of a media space. In K. E. Finn, A. J. Sellen, & S. B. Wilbur (Eds.), *Video-mediated communication* (pp. 245–272). Mahwah, NJ: Lawrence Erlbaum Associates.

Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., & Grinter, R. E. (2005). Quality versus quantity: E-mail centric task management and its relation with overload. *Human-Computer Interaction, 20*, 89–138.

Benford, S., Greenhalgh, C., Reynard, G., Brown, C., & Koleva, B. (1998). Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Transactions on Computer-Human Interaction, 5*, 185–223.

Berners-Lee, T. (1999). *Weaving the web*. New York: Harper Collins.

Bos, N., Gergle, D., Olson, J. S., & Olson, G. M. (2001). *Being there vs. seeing there: trust via video*. Short Paper presented at the Conference on Human Factors in Computing Systems: CHI-2001. Seatle, WA.

Bryant, S. L., Forte, A., & Bruckman, A. (2005). Becoming Wikipedian: Transformation of participation in a collaborative online encyclopedia. *Proceedings of GROUP 2005* (pp. 1–10). New York: ACM.

Bush, V. (1945). As we may think. *Atlantic Monthly, 176*(1), 101–108.

Cadiz, J., Balachandran, A., Sanocki, E., Gupta, A., Grudin, J., & Jancke, G. (2000). *Distance learning through distributed collaborative video viewing.* Paper presented at the CSCW 2000, New York.

Cairncross, F. (1997). *The death of distance: How the communications revolution will change our lives.* Boston: Harvard Business School Press.

Camp, L. J. (2000). *Trust and risk in Internet commerce*. Cambridge, MA: MIT Press.

Carley, K., & Wendt, K. (1991). Electronic mail and scientific communication: A study of the Soar extended research group. *Knowledge: Creation, Diffusion, Utilization, 12*, 406–440.

Carmel, E. (1999). *Global software teams*. Upper Saddle River, NJ: Prentice-Hall.

Cosley, D., Frankowski, D., Kiesler, S., Terveen, L., & Riedl, J. (2005). How oversight improves member-maintained communities. *Proceedings of CHI 2005* (pp. 11–20). New York: ACM.

Daly-Jones, O., Monk, A., & Watts, L. (1998). Some advantages of video conferencing over high-quality audio conferencing: fluency and awareness of attentional focus. *International Journal of Human-Computer Studies, 49*(1), 21–58.

DeSanctis, G., Jackson, B. M., Poole, M. S., & Dickson, G. W. (1996). Infrastructure for telework: Electronic communication at Texaco. *Proceedings of SIGCPR/SIGMIS '96* (pp. 94–102). New York: ACM.

Dourish, P., & Bly, S. (1992). Portholes: Supporting awareness in a distributed work group. *Proceedings of CHI 92* (pp. 541–547). Monterey, CA: ACM Press.

Drolet, A. L. & Morris, M. W. (2000). Rapport in conflict resolution: Accounting for how nonverbal exchange fosters coordination on mutually beneficial settlements to mixed motive conflicts. *Journal of Experimental Social Psychology, 36*(1), 26–50.

Egido, C. (1988). *Video conferencing as a technology to support group work: A review of its failure.* Paper presented at the CSCW '88, New York.

Ehrlich, S. F. (1987). Strategies for encouraging successful adoption of office communication systems. *ACM Transactions on Office Information Systems, 5*, 340–357.

Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware: Some issues and experiences, *CACM, 34*(1), 38–58.

Ellis, C., & Nutt, G. (1980). Office information systems and computer science. *Computing Surveys, 12*(1), 27–60.

Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K, Tang, J., & Welch, B. (1992). LiveBoard: A large interactive display supporting group meetings, presentations, and remote collaboration. *Proceedings of CHI'92* (pp. 599–607). Monterey, CA: ACM Press.

Engelbart, D., & English, W. (1968). A research center for augmenting human intellect. *Proceedings of FJCC, 33*, 395–410.

Federal Communications Commission (2005). *Voice over internet protocol*. Retrieved January 28, 2006, from at http://www.fcc.gov/voip/

Federal Trade Commission (2005, November). *Email address harvesting and the effectiveness of anti-spam filters*. Report by Federal Trade Commission, Division of Marketing Practices. Washington, DC.

Finholt, T. A. (2002). Collaboratories. In B. Cronin (Ed.), *Annual Review of Information Science and Technology*.

Finholt, T. A., & Olson, G. M. (1997). From laboratories to collaboratories: A new organizational form for scientific collaboration. *Psychological Science, 8*, 28–36.

Finholt, T., Sproull, L., & Kiesler, S. (1990). Communication and performance in ad hoc task groups. In J. Galegher, R. Kraut, & C. Egido (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work* (pp. 291–325). Hillsdale, NJ: Lawrence Erlbaum Associates.

Finn, K., Sellen, A., & Wilbur, S. (Eds.) (1997). *Video-mediated communication*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Fish, R. S., Kraut, R. E., Root, R. W., & Rice, R. E. (1993). Video as a technology for informal communication. *Communications of the ACM, 36*(1), 48–61.

Foster, I., & Kesselman, C. (2004). *The Grid: Blueprint for a new computing infrastructure* (2nd ed.). San Francisco: Morgan Kaufmann.

Fussell, S. R., Kraut, R. E., & Siegel, J. (2000). *Coordination of communication: Effects of shared visual context on collaborative work.* Paper presented at the CSCW 2000, New York.

Gale, C., & Monk, A. (2000). Where am I looking? The accuracy of video-mediated gaze awareness. *Perception & Psychophysics, 62*, 586–595.

Giles, J. (2005). Internet encyclopaedias go head to head. *Nature, 438*, 900–901.

Godefroid, P., Herbsleb, J. D., Jagadeesan, L. J., & Li, D. (2000). Ensuring privacy in presence awareness systems: An automated verification approach. *Proceedings of CSCW 2000* (pp. 59–68). New York: ACM.

Grayson, D. M., & Monk, A. (2003). Are you looking at me? Eye contact and desktop video conferencing. *ACM Transactions on Computer-Human Interaction, 10*(3), 221–243.

Greif, I. (Ed.) (1988). *Computer-supported cooperative work: A book of readings*. San Mateo, CA: Morgan Kaufmann.

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *Proceedings of the Con-*

ference on Computer Supported Cooperative Work (pp. 85–93). Portland, OR: ACM Press.

Grudin, J. (1994). Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM, 37*(1), 92–105.

Grudin, J., & Palen, L. (1995). Why groupware succeeds: Discretion or mandate? *Proceedings of the European Computer Supported Cooperative Work* (pp. 263–278). Stockholm, Sweden: Springer.

Gutwin, C., & Greenberg, S. (1999). The effects of workspace awareness support on the usability of real-time distributed groupware. *ACM Transactions on Computer-Human Interaction, 6*, 243–281.

Hall, E. T. (1982). *The Hidden Dimension.* New York: Anchor Doubleday Books.

Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2000). Distance, dependencies, and delay in a global collaboration. *Proceedings of CSCW 2000* (pp. 319–328). New York: ACM.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of CSCW 2000* (pp. 241–250). New York: ACM.

Hindmarsh, J., Fraser, M., Heath, C., Benford, S., & Greenhalgh, C. (1998). Fragmented Interaction: Establishing Mutual Orientation in Virtual Environments, *Proceedings of Conference on Computer-Supported Cooperative Work* (pp. 217–226). Portland, OR: ACM Press.

Hollingshead, A. B., McGrath, J. E., & O'Connor, K. M. (1993). Group performance and communication technology: A longitudinal study of computer-mediated versus face-to-face work. *Small Group Research, 24*, 307–333.

Huang, W., Olson, J. S., & Olson, G. M. (2002). *Camera angle affects dominance in video-mediated communication.* Paper presented at the CHI 2002, New York.

Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D. J., & Kamm, C. (2002). *The character, functions, and styles of instant messaging in the workplace.* Paper presented at the CSCW 2002, New York.

Jensen, C., Farnham, S. D., Drucker, S. M., & Kollock, P. (2000). The effect of communication modality on cooperation in on-line environments. *Proceedings of CHI '2000.* (pp. 470–477). New York: ACM Press.

Karsenty, L. (1999). Cooperative work and shared visual context: An empirical study of comprehension problems in side-by-side and remote help dialogues. *Human-Computer Interaction, 14*, 283–315.

Kendon, A. (1967). Some functions of gaze direction in social interaction. *Acta Psychologia, 26*, 22–63.

Kiesler, S., & Cummings, J. N. (2002). What do we know about proximity and distance in work groups? A legacy of research. In P. J. Hinds & S. Kiesler (Eds.), *Distributed work* (pp. 57–80). Cambridge, MA: MIT Press.

Kiesler, S., Sproull, L, & Waters, K. (1996). Prisoner's dilemma experiment on cooperation with people and human-like computers. *Journal of Personality and Social Psychology, 70*(1), 47–65.

Koschmann, T. (1996). *CSCL: Theory and practice of an emerging paradigm*. Lawrence Erlbaum Associates. Hillsdale, NJ.

Koschmann, T., Hall, R., & Miyake, N. (2002). *CSCW 2: Carrying forward the conversation*. Lawrence Erlbaum Associates. Hillsdale, NJ.

Koschmann, T., Suthers, D. D., & Chan, T. (2005). *Computer supported collaborative learning 2005: The next 10 years!* Lawrence Erlbaum Associates. Hillsdale, NJ.

Kraemer, K. L., & Pinsonneault, A. (1990). Technology and groups: Assessments of empirical research. In J. Galegher, R Kraut, & C. Egido (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work* (pp. 373–405). Hillsdale, NJ: Lawrence Erlbaum Associates.

Krauss, R. M., & Bricker, P. D. (1966). Effects of transmission delay and access delay on the efficiency of verbal communication. *Journal of the Acoustical Society, 41*, 286–292.

Kraut, R. E., Fussell, S. R., & Siegel, J. (2003). Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction, 18*(1–2), 13–39.

Kraut, R. E., Gergle, D., & Fussell, S. R. (2002). *The use of visual information in shared visual spaces: Informing the development of virtual co-presence.* Paper presented at the CSCW 2002, New York.

Kraut, R., Kiesler, S., Boneva, B., Cummings, J., Helgeson, V. & Crawford, A. (2002). Internet paradox revisited. *Journal of Social Issues, 58*(1), 49–74.

Kraut, R., Patterson, M., Lundmark, V., Kiesler, S., Mukopadhyay, T., & Scherlis, W. (1998). Internet paradox: A social technology that reduces social involvement and psychological well-being. *American Psychologist, 53*, 1071–1031.

Kraut, R. E., Sunder, S., Telang, R., & Morris, J. (2005). Pricing electronic mail to solve the problem of spam. *Human-Computer Interaction, 20*, 195–223.

Lange, B. M. (1992). Electronic group calendaring: Experiences and expectations. In D. Coleman (Ed.) *Groupware* (pp. 428–432). San Mateo, CA: Morgan Kaufmann.

Longstaff, T. A., Ellis, J. T., Hernan, S. V., Lipson, H. F., McMillan, R. D., Pesanti, L. H., & Simmel, D. (1997). Security on the Internet. In *The Froehlich/Kent Encyclopedia of Telecommunications* (Vol. 15, pp. 231–255). New York: Marcel Dekker.

Luff, P., Heath, C., Kuzuoka, H., Hindmarsh, J., & Oyama, S. (2003). Fractured ecologies: Creating environments for collaboration. *Human-Computer Interaction, 18*(1–2), 51–84.

Mackay, W. E. (1989). Diversity in the use of electronic mail: A preliminary inquiry. *ACM Transactions on Office Information Systems, 6*, 380–397.

Malone, T. W., Grant, K. R., Lai, K. Y., Rao, R., & Rosenblitt, D. A. (1989). The information lens: An intelligent system for information sharing and coordination. In M. H. Olson (Ed.). *Technological support for work group collaboration* (pp. 65–88). Hillsdale, NJ: Lawrence Erlbaum Associates.

Markus, M. L. (1983). *Systems in Organization: Bugs and Features.* San Jose, CA: Pitman.

Markus, M. L., & Connolly, T. (1990). Why CSCW applications fail: Problems in the adoption of interdependent work tools. *Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 371–380). Los Angeles, CA: ACM Press.

McDaniel, S. E., Olson, G. M., & Magee, J. S. (1996). Identifying and analyzing multiple threads in computer-mediated and face-to-face conversations. *Proceeding of the ACM Conference on Computer Supported Cooperative Work* (pp. 39–47). Cambridge, MA: ACM Press

McLeod, P. L. (1992). An assessment of the experimental literature on electronic support of group work: Results of a meta-analysis. *Human-Computer Interaction, 7*, 257–280.

Monk, A., & Gale, C. (2002). A look is worth a thousand words: Full gaze awareness in video-mediated communication. *Discourse Processes, 33*(3), 257–278.

Moran, T. P., Chiu, P., Harrison, S., Kurtenbach, G., Minneman, S., & van Melle, W. (1996). Evolutionary engagement in an ongoing collaborative work process: A case study. *Proceeding of the ACM Conference on Computer Supported Cooperative Work* (pp. 150–159). Cambridge, MA: ACM Press.

Mosier, J. N., & Tammaro, S. G. (1997). When are group scheduling tools useful? *Computer Supported Cooperative Work, 6*, 53–70.

Muller, M. J., Raven, M. E., Kogan, S., Millen, D. R., & Carey, K. (2003). *Introducing chat into business organizations: Toward an instant messaging maturity model.* Paper presented at the GROUP '03, New York.

Nardi, B. A., Schwarz, H., Kuchinsky, A., Leichner, R., Whittaker, S., & Sclabassi, R. (1993). *Turning away from talking heads: the use of video-as-data in neurosurgery.* Paper presented at the CHI 93, New York.

Nardi, B. A., Whittaker, S., & Bradner, E. (2000). Interaction and outeraction: Instant messaging in action. *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (pp. 79–88). Philadelphia, PA: ACM Press

Nardi, B. A., Schiano, D. J., Gumbrecht, M., & Swartz, L. (2004). Why we blog? *Communications of the ACM, 47*(12), 41–46.

National Research Council. (1993). *National collaboratories: Applying information technology for scientific research*. Washington, D.C.: National Academy Press.

Nohria, N., & Eccles, R. G. (Eds.). (1992). *Networks and organizations: Structure, form, and action.* Boston: Harvard Business School Press.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., Vogel, D. R., & George, J. F. (1991). Electronic meeting systems to support group work. *Communications of the ACM, 34*(7), 40–61.

O'Conaill, B., Whittaker, S., & Wilbur, S. (1993). Conversations over videoconferences: An evaluation of the spoken aspects of video mediated communication. *Human-Computer Interaction, 8*, 389–428.

O'Hara-Devereaux, M., & Johansen, R. (1994). *Global work: Bridging distance, culture & time*. San Francisco: Jossey-Bass.

Okada, K., Maeda, F., Ichicawaa, Y., & Matsushita, Y. (1994). *Multiparty videoconferencing at virtual social distance: MAJIC design.* Paper presented at the CSCW 94, New York.

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human-Computer Interaction, 15*, 139–179.

Olson, G. M., Atkins, D., Clauer, R., Weymouth, T., Prakash, A., Finholt, T., Jahanian, F., & Rasmussen, C. (2001). Technology to support distributed team science: The first phase of the Upper Atmospheric Research Collaboratory (UARC) In G. M. Olson, T. Malone, & J. Smith (Eds.), *Coordination theory and collaboration technology* (pp. 761–783). Hillsdale, NJ: Lawrence Erlbaum Associates.

Olson, G. M., Olson, J. S., Bos, N., & the SOC Data Team (2004). International collaborative science on the net. In W. Blanpied (Ed.), *Proceedings of the Trilateral Seminar on Science, Society and the Internet* (pp. 65–77). Arlington, VA: George Mason Univeristy.

Olson, J. S., Olson, G. M., & Meader, D.K. (1995). What mix of video and audio is useful for remote real-time work? *Proceedings of CHI '95* (pp. 362–368). Denver, CO: ACM Press.

Olson, J. S., Olson, G. M., Storrøsten, M., & Carter, M. (1993). Group work close up: A comparison of the group design process with and without a simple group editor. *ACM Transactions on Information Systems, 11*, 321–348.

Olson, J. S., & Teasley, S. (1996). Groupware in the wild: Lessons learned from a year of virtual collocation. *Proceeding of the ACM Conference on Computer Supported Cooperative Work* (pp. 419–427). Cambridge, MA: ACM Press.

Olson, M. H., & Bly, S. A. (1991). The Portland experience: A report on a distributed research group. *International Journal of Man-Machine Studies, 34*, 211–228.

Orlikowski, W. J., & Gash, D. C. (1994). Technological frames: Making sense of information technology in organizations. *ACM Transactions on Information Systems, 12*, 174–207.

Palen, L., & Grudin, J. (2002). Discretionary adoption of group support software: lessons from calendar applications. In B. E. Munkvold (Ed.), *Implementing collaboration technologies in industry* (pp. 159–180). Springer-Verlag. London.

Park, K. S., Kapoor, A., & Leigh, J. (2000). Lessons learned from employing multiple perspective in a collaborative virtual environment for visualizing scientific data. *Proceedings of ACM CVE'2000 Conference on Collaborative Virtual Environments* (pp. 73–82). San Francisco, CA: ACM Press.

Prinz, W., & Kolvenbach, S. (1996). Support for workflows in a ministerial environment. *Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 199–208). Cambridge, MA: ACM Press.

Putnam, R. D. (2000). *Bowling alone: The collapse and revival of American community*. New York: Simon & Schuster.

Resnick, P. (2002). Beyond bowling together: SocioTechnical capital. In J. M. Carroll (Ed.), *Human-computer interaction in the new millennium* (pp. 647–672). New York: ACM Press.

Resnick, P., & Varian, H. R. (Eds.) (1997). Special section: Recommender systems. *Communications of the ACM, 40*(3), 56–89.

Rocco, E. (1998) Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact. *Proceedings of CHI'98* (pp. 496–502). Los Angeles, CA: ACM Press.

Rocco, E., Finholt, T., Hofer, E. C., & Herbsleb (2000). *Designing as if trust mattered*. (CREW Technical Report). University of Michigan, Ann Arbor.

Ruhleder, K., & Jordan, B. (2001). Co-constructing non-mutual realities: Delay-generated trouble in distributed interaction. *Computer Supported Cooperative Work, 10*(1), 113–138.

Satzinger, J., & Olfman, L. (1992). A research program to assess user perceptions of group work support. *Proceeding of CHI'92* (pp. 99–106). Monterey, CA: ACM Press.

Schafer, J. B., Konstan, J., & Riedl, J., (2001). Electronic commerce recommender applications. *Journal of Data Mining and Knowledge Discovery, 5*(1/2), 115–152.

Schatz, B. R., & Hardin, J. B. (1994). NCSA Mosaic and the World Wide Web: Global hypermedia protocols for the internet. *Science, 265*, 895–901.

Schmandt, C., & Marti, S. (2005). Active Messenger: E-mail filtering and delivery in a heterogeneous network. *Human-Computer Interaction, 20*, 163–194.

Short, J., Williams, E., & Christie, B. (1976). *The social psychology of telecommunications*. New York: Wiley.

Siegel, J., Dubrovsky, V., Kiesler, S., & McGuire, T. W. (1986). Group processes in computer-mediated communication. *Organizational Behavior and Human Decision Processes, 37*(2), 157–187.

Singhal, S., & Zyda, M. (1999). *Networked virtual environments: design and implementation*. New York: Addison-Wesley.

Slavin, R. E. (1994). *Cooperative learning: Theory, research, and practice* (2nd ed.). Boston: Allyn & Bacon.

Sproull, L., & Kiesler, S. (1991). *Connections: New ways of working in the networked organization.* Cambridge, MA: MIT Press.

Starner, T., & Rhodes, B. (2004). Wearable computer. In W. S. Bainbridge (Ed.), *Berkshire encyclopedia of human-computer interaction* (Vol. 2, pp. 797–802). Great Barrington, MA.: Berkshire Publishing Group.

Straus, S. G. (1996). Getting a clue: The effects of communication media and information distribution on participation and performance in computer-mediated and face-to-face groups. *Small Group Research, 1*, 115–142.

Straus, S. G. (1997). Technology, group process, and group outcomes: Testing the connections in computer-mediated and face-to-face groups. *Human-Computer Interaction, 12*(3), 227–266.

Straus, S. G., & McGrath, J. E. (1994). Does the medium matter: The interaction of task and technology on group performance and member reactions. *Journal of Applied Psychology, 79*, 87–97.

Tanenbaum, A. S. (2003). *Computer networks* (4th ed.). Upper Saddle River, NJ: Prentice Hall PTR.

Tang, J. C., & Isaacs, E. (1993). Why do users like video? *Computer Supported Cooperative Work, 1*(3), 163–196.

Tang, J. C., Yankelovich, N., Begole, J., van Kleek, M., Li, F., & Bhalodia, J. (2001). ConNexus to Awarenex: Extending awareness to mobile users. *Proceedings of CHI 2001* (pp. 221–228). New York: ACM.

Teasley, S. D., Covi, L. A., Krishnan, M .S., & Olson, J. S. (2002). Rapid software development through team collocation. *IEEE Transactions on Software Egnineering, 28*, 671–683.

Veinott, E., Olson, J. S., Olson, G. M., & Fu, X. (1999). Video helps remote work: Speakers who need to negotiate common ground benefit from seeing each other. *Proceedings of the Conference on Computer-Human Interaction, CHI'99* (pp. 302–309).Pittsburgh, PA: ACM Press.

Vertegaal, R. (1999). *The GAZE groupware system: Mediating joint attention in multiparty communication and collaboration.* Paper presented at the CHI 99, New York.

Vertegaal, R., Slagter, R., van der Veer, G., & Nijholt, A. (2001). *Eye gaze patterns in conversations: There is more to conversational agents than meets the eye.* Paper presented at the CHI 2001, New York.

Viegas, F. B., Wattenberg, M., & Dave, K. (2004). Studying cooperation and conflict between authors with history flow visualizations. *Proceedings of CHI 2004* (pp. 575–582). New York: ACM.

Wellman, B. (2001). *Design considerations for social networkware: Little boxes, glocalization, and networked individualism*. Draft ms.

Whittaker, S. (2005). Supporting collaborative task management in e-mail. *Human-Computer Interaction, 20*, 49–88.

Whittaker, S., Bellotti, V., & Moody, P. (2005). Introduction to this special issue on revisiting and reinventing e-mail. *Human-Computer Interaction, 10*, 1–9.

Whittaker, S., Geelhoed, E., & Robinson, E. (1993). Shared workspaces: How do they work and when are they useful? *International Journal of Man-Machine Studies, 39*(5), 813–842.

Whittaker, S., & Sidner, C. (1996). Email overload: Exploring personal information management of email. *Proceeding of CHI'96* (pp. 276–283). Vancouver, BC: ACM Press.

Williams, E. (1977). Experimental comparisons of face-to-face and mediated communication: A review. *Psychological Bulletin, 84*, 963–976.

Winograd, T. (1988). A language/action perspective on the design of cooperative work. *Human Computer Interaction, 3*, 3–30.

Yang, H., & Olson, G. M. (2002). Exploring collaborative navigation: the effect of perspectives on group performance. *Proceedings of CVE '02* (pp. 135–142). New York: ACM Press.

Zheng, J., Bos, N., Olson, J. S., Gergle, D. & Olson, G. M. (2001). Trust without touch: Jump-start trust with social chat. *Paper presented at the Conference on Human Factors in Computing Systems CHI-01.* Seatle, WA: ACM Press

·28·

HCI AND THE WEB

Helen Ashman, Tim Brailsford, Gary Burnett, Jim Goulding, Adam Moore, and Craig Stewart
University of Nottingham

Mark Truran
University of Teesside

Introduction **560**
What Makes the Web Hard to Use? **560**
Browsing and Linking: "What's Wrong With the World Wide Web?" Revisited 560
Finding Things (i): Search and Query on the web 561
Finding Things (ii): Relevance 562
User Interface Issues 562
Context of Use 562
Navigation Issues 563
Summary 563
Browsing and Linking **563**
Broken and Misdirected Links 564
Personalizing Links 564
Searching and Querying **565**
Query Formulation 565
Relevance Feedback 566
The Results List 566
Community-Based Ranking Algorithms 566
Improved Visual Interfaces 567
Document Clustering 567
Summary 567
Personalization, Portals and Communities **567**
Personalization 568
Adaptive and Adaptable Systems 568
Need or Desire? 569
Methods and Models of Adaptation 569
User Modeling 569
Authoring 570
Portals 570
Communities 570
Conclusions **571**
References **571**

INTRODUCTION

Pioneering research work in the early 1980s introduced and defined the concept of usability as critical to the success of interactive products and systems (Eason, 1984). From the 1990s there has been a developing relationship between usability and the web, such that for many people the two terms are inextricably linked. Specifically, not only does the web demand good usability, it can be argued that usability necessitates the web.

There are five key reasons why usability and the web have become so closely associated with each other:

1. Web users are ubiquitous. For many countries, the people who access the web are approaching the population as a whole. For instance, within Great Britain 55% of households can now access the Internet from home.[1] For some countries (i.e., United States, Denmark, Australia), it has been estimated that over 70% of the population are regularly using the web.[2] Such diversity raises a large number of issues related to interdependent variables such as age, gender, culture, disability, language abilities, computer skills/knowledge, domain skills/knowledge, and so on. In particular, considerable ongoing research considers the *accessibility* of the web; that is, websites can be designed to account for the needs of as many people as possible (i.e., those with visual impairments) (Abascal & Nicolle, 2001).
2. Web users are largely discretionary users. Apart from the specific case of the work context and company intranets, web users generally do not have to use a particular site (or even the web at all) to achieve their goals. They have alternatives available to them (i.e., visiting another website, making a phone call, visiting a shop), and if they experience usability problems, they do not necessarily have to struggle with or adapt to the poor interface. They are empowered to explore existing options.
3. Web usability problems have a clear relationship with sales. For websites aiming to sell products or services, poor usability directly impacts sales. If a user cannot find the product or relevant information, he or she is unlikely to continue in the transaction. In this case, it has been noted that usability is affecting the experience of a product *prior* to purchase (Nielson, 2000). Consequently, the website information space must consider many of the issues (i.e., navigation, layout of items, labelling) that are relevant to the traditional design of physical space for shops.
4. The web is evolving at a rapid pace. Technical characteristics are continually changing in response to new application/task areas, facilitated by computer processing power and communication speeds. This has an immediate impact on the functionality available to web users (i.e., animations, videos), developments in user interface design (i.e., clickable items, mouse-over navigation), as well as tools (i.e., cookies, plug-ins). In contrast, users' skills, knowledge, and expectations are considerably slower to evolve, leading to an inevitable gulf between users (particularly infrequent users) and the web.
5. Website technical development is easy. It is simple to have a presence on the web when compared with the resources necessary for traditional product-development processes. To generate a website requires few technical skills, and one can do so with no programming experience. This has contributed to the vast number of websites[3] ultimately adding to the complications of navigation, whilst emphasising the importance of usability as a differentiating factor.

In this chapter we first consider what is it that makes the web hard to use. We look specifically at browsing and linking, finding things with search and query, increasing the relevance of things we find with personalization, and user interface issues. The subsequent sections look at some solutions to these problems.

WHAT MAKES THE WEB HARD TO USE?

As successful as the web is for delivering information globally and rapidly, many problems remain that make it an unpleasant or unproductive experience for some users, or even impossible for other users.

Much of the difficulty in using the web lies in the vast quantity of information available—browsing and searching becomes increasingly difficult and imprecise in its results. Then having located what one wants, technical flaws, such as broken links or browser incompatibilities, often render the information unreachable or unreadable. Most importantly, web pages or the browser interface itself can give a poor presentation, and for some classes of users, make the information inaccessible.

Browsing and Linking: "What's Wrong With the World Wide Web?" Revisited

Almost a decade ago, the difficulties in using the web were explored in-depth, and presented a vision of "fourth-generation hypermedia" (Bieber, Vitali, Ashman, Balasubramanian, & Oinas-Kukkonen, 1997). In particular, the importance of hypermedia in the structuring of information was described:

> hypermedia provides contextual, navigational access for viewing information and . . . represents knowledge in a form relatively close to the cognitive organizational structures that people use. Thus hypermedia supports understanding.

The judicious use of hypermedia enhances a user's comprehension. When that use is tailored specifically to the user's requirements, i.e., it is *personalized*, the user's comprehension and productivity are maximized.

[1]Figures for July 2005 from web.statistics.gov.uk

[2]Figures for 2005 from www.internetworldstats.com

[3]Estimated to be over 9 million in 2002—www.oclc.org

The authors went on to define a table of desired hypermedia features (Bieber et al., 1997). Of particular relevance to HCI and the web are the personalization of links by annotation, computation, and overviews.

Surprisingly, nearly a decade later, many of these desired features are still lacking from the mainstream web applications. In particular the personalization of links and content is still largely unachievable without specialist tools. In fact, some backsliding on this is evident over the past decade, as the mid-1990s Mosaic browser supported the creation of annotations at both personal and group level and, more importantly, offered this service by default in a standard browser.[4] Mosaic development, however, was discontinued when Netscape became widespread. W3C's Annotea project[5] offers the same essential functionality but not within a standard browser and not without setting up effort from a technically literate user—annotations are by no means a standard or easily accessible web service.

Creating personalized annotations is challenging enough, but the creation of one's own links over third-party data sources is essentially unknown. This is not due to any shortcomings in the technology; as in fact many solutions have been available within the research community for some time (see "Browsing and Linking"). The earliest plans for web browsers, dating back to 1991 (Cailliau & Ashman, 1999), anticipated a "writeable web" where personalized links and annotations would be easily supported, yet this has not become mainstream web technology.

Another key aspect of personalized links which is inevitably overlooked is that personal links are *private* links. If one creates a web page, complete with links, then those links are necessarily as available as the content. This is not always desirable, especially if links record one's associations and collected thoughts, and the intellectual property represented by those associations has value. The technology for personalized links gives the user private links, over public documents, in much the same way that users' bookmarks are private. For privacy reasons alone, the technology for personalized links should be requisite for all mainstream browsers.

Other desired hypermedia features, such as local overviews, are provided not as a standard web service but as a site-specific courtesy page optionally supplied by site managers. Global overviews seem an increasingly distant feature, as the vast complexity of the web makes the usual, graph-based representations completely unworkable, both computationally and visually. The earlier paper noted that:

> Web browsers have no inherent way of presenting the structure and interrelationships of data of any sort. For example, there is no way to visualize even the simple interrelationships of web documents, such as "Where can I go from here?" or "Which documents point to this document?" The reader has no idea of the position of a given document within the corpora unless an author explicitly embeds such details. (Bieber et al., 1997)

So why is it that so little has seemingly been done to address these outstanding problems? It could be because the technology is too inaccessible to the nonspecialist user and that the obvious upfront costs of assimilating the technology outweigh the perceived benefits. It still seems to be true that:

> experienced readers learn to get by with the available functionality. However, the use of web technology is in part determined by its capabilities. Readers learn to make do with the available tools instead of demanding better tools, perhaps because they are not aware of how better tools might help them. (Bieber et al., 1997)

Only when we make the web more usable to the nonspecialist user will the use of all but the most basic tools will become commonplace, resulting in higher productivity and better assimilation of information in the user.

So far, we have seen the web does not offer widespread support for personalization of links and other browsing activities. We now look at the other major means of web information access: searching and querying.

Finding Things (i): Search and Query on the Web

One recent estimate has placed the size of the World Wide Web at around 11.5 billion pages (Gulli & Signorini, 2005). This figure, representing nearly 1.8 web pages per person alive today,[6] relates only to static or fixed web pages. It has been suggested that if this survey was extended to include more ephemeral information (i.e., individual auction items or ongoing news stories), then the correct figure is in fact 500 times larger (Bergman, 2001).

With this wealth of information, the web would be untenable without mechanisms to assist navigation and file location. The most common web tool in use today is the search engine. To use a search engine the user must submit a series of terms known as a query. This query in some way formulates and embodies what the user of the search system would like to retrieve information about. The usual output produced by the search engine is a small set of links to a selection of web documents extracted from the billions available. These web pages represent the search engine's response to the user's information need, each document having been determined by the system as probably relevant to the query.

Presented with a set of results, the user selects a web page. If the document fails to meet the user's requirement, they return to the search engine for another result, or modify the query in light of previous results. This continues until the user's need is satisfied or their commitment to the search wanes (Harman, 1992).

From the perspective of a typical user, this has two main problem areas:

1. the confusion and uncertainty surrounding query formulation
2. the impenetrability of seemingly endless results

In "Searching and Querying" we discuss these obstacles and review a range of solutions.

---

[4]In 1994 the Mosaic browser was the "killer application" for the Internet, being the first major web browser, freely available from NCSA. Its developers later spun off the Netscape company.

[5]http://www.w3.org/2001/Annotea/

[6] http://www.geohive.com approximates the population of the Earth at 6 483 981 671 (5th January 2005).

Finding Things (ii): Relevance

While browsing and search-and-query are excellent tools for finding information on the web, they remain generic functions whose behavior is the same, regardless of the context. What remains to be considered now is how to further filter the mass of information on the web according to its relevance, not to a query, but to the individual user or to a community of users.

Most information systems are designed for a hypothetical "average" user. This "one-size-fits-all" approach ignores diversity in cultural and educational backgrounds, abilities, objectives and aspirations. An information system with a single user interface for all users is conceptually the same a car manufacturer selling a car in only a single color—"any color so long as it's black".[7]

One solution is to build personalizable information systems, delivering content specific to requirements of different users. Without such systems, effective universal accessibility to information cannot be achieved. With it, every computer user and special interest group will have personally tailored access to information sources.

Personalization can take many forms, and can involve the tailoring of hypertext links (as discussed in "What Makes the Web Hard to Use"), the tailoring of presentation (often an accessibility issue, see "User Interface Issues") and the tailoring of content.

Content personalization has many benefits, for not just users, either as individuals or within a community, but also for the providers of information.

For individuals, the tailoring of content can reflect a wide variety of requirements, but is always aimed at providing information most pertinent to that user in their current context. In e-learning applications for example, more challenging lessons are not served to the user until mastery of prerequisite material is achieved. In e-commerce, a simple form of user consensus (see "Document Clustering") underlies a recommender system that personalizes suggestions for further purchases based on the current users' purchasing history similarity to that of other users.

For groups, personalization of content could help create communities of common interest amongst otherwise disparate users. Personalization can also support existing communities of users, by creating portals for accessing materials which are "personalized" to the interests of the group or community.

Personalization of information can also improve the whole computer use experience for the two major groups of underrepresented users: (a) non-American/English cultural and language groups, and (b) people with special information needs. The web may have been "invented" by Europeans (Cailliau et al., 1999), but its subsequent development from 1993 was U.S.-based and has a distinctly American culture, including a predominance of the English language, linguistic form and alphabets.[8]

For information providers and publishers, personalization could create a delivery of information as suitable as possible to each individual user or special-interest group. In the normal publishing model, it is not economically feasible for publishers or information owners to publish information in a multitude of different forms, but with personalization and adaptation (see "Personalization"), it would be possible to do the personalization "on the fly" at no additional cost to the publishers, bringing enormous economic advantage to publishers using the technology.

This is essentially the same principle encountered in teaching lectures present the information in a fairly general fashion, while tutorials or other small-group teaching give the students a chance to specify exactly what they need to know and to be given examples most helpful to them. In fact, the pressure on teaching resources has partly motivated adaptation and personalization in e-learning, and similarly it is the lack of resources preventing publishers from delivering their materials tailored to special-interest groups and to the individual. With e-learning, adaptation and personalization have arisen because teachers must offer personalized tutoring but are increasingly short of resources. With e-commerce, multicultural groups and special needs groups, adaptation and personalization can also be applied to offer an equally valuable specialisation of information that is just not otherwise available.

User Interface Issues

We have just seen how awareness of context played an important part in the personalization of content and links. However context also needs to be considered in any environment, not just for personalization purposes.

Context of Use

When investigating the usability of web user interfaces, one must first consider the overriding issue of context. Context of use is seen as a critical constituent of usability, defined by ISO (1998) to "consist of the users, tasks, and equipment (hardware, software, and materials) and the physical and social environments in which a product is used." With respect to the web, the user, the task, and the environment, issues are constantly evolving, leading to new challenges for HCI researchers and practitioners. For instance, the traditional environment for web users would have been the workplace, but statistics show that this is no longer the single situation for web use. Eighty-eight percent of UK web users access the Internet primarily from home.[9] Also, an increasing number of people now access the Internet through mobile devices (i.e., laptops, phones, PDAs). In these situations, designers must consider a much wider range of physical environment factors, for instance, varying lighting, noise and thermal conditions, as well as other tasks that users may simultaneously carry out.

---

[7]Henry Ford is credited with saying "You can paint it any color, so long as it's black," but the Model T eventually appeared in many colors. Customer demand can motivate personalization.

[8]The American culture and set of assumptions is reflected in not only issues such as inadvertently rendering Arabic script backwards (left to right) because browsers render left to right by default, but also in American spelling of HTML tags, such as "color" and "gray."

[9]Figures for July 2005 from www.statistics.gov.uk

For example, there are widespread concerns regarding the use of Internet services within road-based vehicles (Lai, Cheng, Green, & Tsimhoni, 2002; Burnett, Summerskill, & Porter, 2004). Access to the web while driving may provide a range of tangible benefits to drivers, some driving-related (i.e., pre-booking a parking space, accessing real-time traffic information), and others oriented toward productivity or entertainment needs (i.e., viewing the latest information on stocks, downloading MP3 files). Traditional access to the web through desktop computers has utilized user interface paradigms that are both highly visual and manual (i.e., scanning a page for a link and then using a mouse to point and then click on the link). Such activities are in clear conflict with the safety/time-critical task of driving, which places heavy demands on the visual modality, while requiring continuous manual responses (i.e., turning the steering wheel). Clearly, designers need to establish fundamentally new interaction styles for use in a driving context of use (i.e., speech recognition, voice output, haptic interfaces; see for example Iwata et al. in this book). As part of this process, designers of interfaces for in-car Internet services must consider whether drivers should be given access to functionality while the vehicle is in motion. Considerable ongoing research is being conducted to determine what constitutes an overly distracting interface (Lai et al., 2002).

Navigation Issues

There is a common assertion that the two biggest difficulties facing web users are download times and navigation (Nielson, 2000; McCracken & Wolfe, 2004). Download times are largely governed by the technical capabilities of Internet connections (which have increased markedly over recent years), together with clear design variables (i.e., the size of individual website pages). Navigation issues are considerably more complex. According to one diary-based study (Lazar, Bessiere, Ceaparu, Robinson, & Shneiderman, 2003), between one third and one half of time spent using a computer is unproductive, a situation predominately attributed to problems in web navigation.

In analyzing the navigation problem, we must first consider what is meant by a navigation task. A CHI workshop on this topic from 1997 (Jul & Furnas, 1997) commented that a difficulty with research in the area was that authors tended to define navigation in varying ways. A broad view (adapted from the CHI workshop) is taken here, in which navigation can be said to involve the following:

- Planning "routes." When people navigate through space (whether real or electronic), they must first consider their overall strategy; that is, what methods will be appropriate in the current situation. For the web, a range of methods exist that aim to assist the user when deciding how to navigate across the web (i.e., search engines, directories, URLs) and/or within specific sites (i.e., site maps, navigation menus, links). Many people find it difficult to generate a suitable plan for a range of reasons, either concerning basic cognitive limitations (such as remembering URLs), a lack of knowledge (choosing appropriate search terms, misunderstanding Boolean logic), or because methods are poorly implemented (i.e., confusing layouts for site maps).
- Following "routes." Once a high-level plan exists, people need to execute the subsequent point-by-point decisions, necessitating a virtual form of locomotion through the information space. In this stage, typical problems facing the web user often relate to the design of linking mechanisms between pages, for example, ambiguous link labelling, unclear graphics or icons, relevant information appearing off-screen, the need to visually scan large numbers of links, and so on.
- Orienting within the "space." The "where am I?" problem of the web is perhaps the one most discussed and researched (Otter & Johnson, 2000). For optimum navigation performance and confidence, people need to have a sense of their current location in relation to their surroundings (i.e., their final destination, their start point, and other key "landmarks," such as a home page). Orientation difficulties are often compounded on the web because users are "dropped" into a specific location when they use search engines and bookmarks.
- Learning the "space." Repeated exposure to any large-scale environment (whether it be real or electronic) will lead to a deepening knowledge of objects within the space (i.e., particular pages), as well as an understanding of the various relationships between the objects (i.e., how pages follow each other, the overall structure of a site). These specific cases of mental models are commonly known as cognitive maps, and facilitate fast and accurate navigation performance. websites with poor differentiation (i.e., all pages appearing to be similar), low visual access (i.e., difficult to see where one can go next) and high path complexity (i.e., many links on a page) will all contribute to a poorly formed cognitive map (Kim & Hurtle, 1995).

Summary

In the subsequent sections, we turn to some solutions for the problems outlined here, looking first at browsing and linking ("Browsing and Linking"), then turning to search ("Searching and Querying") and personalization ("Personalization, Portals, and Communities").

BROWSING AND LINKING

The fundamental mechanism for viewing information on the web is by browsing implemented by hypermedia links.[10] However, this core capability still encounters technical problems that interrupt the user's browsing, such as difficulties in finding relevant material and limitations in the way relevant materials can be presented.

[10]The very names of the two most important elements of the web infrastructure, *hypertext* transfer protocol (http) and *HyperText* Markup Language (HTML) indicate this clearly.

In this section, we look at some solutions, including automatic management of broken links, the easy personalization of links, and how links can enable different perspectives on the same data.

Broken and Misdirected Links

Broken links have frustrated users not only of the web, but also of earlier hypertext systems, with some of the first work on link integrity motivated the open hypermedia systems' principle of externalizing links, rather than embedding them into the data (Davis, 1995).

Externalizing links implies that links are stored separately from the data being linked, so that reconciliation of the link to its referent was required, usually immediately before use, as a "late binding" of links (Brailsford, 1999). However any changes in the data meant that links could be either displaced, pointing to the wrong place in a document (part-of-file error, Davis, 1998), or completely invalidated, with no document now known for the link to apply to (a whole-file error, Davis, 1998).

HTML links suffer from a similar problem, despite being embedded in the data, since it is not only the source of the link (the "from" part of it) that must be reconciled to its referent, but also the destination (the "to" part). It is in the latter that HTML links are frequently wrong, with an estimated 23% of web links being broken within a year (Lawrence, Pennock, Flake, Krovetz, Coetzee, Glover, Nielsen, Kruger, & Giles, 2001).

Broken links can be frustrating for the user, resulting in loss of interest in the website. Fortunately, this is one area that has seen ongoing improvement.

The solutions to broken links can be characterised as being *preventative* (creating infrastructure or procedures that avoid broken links), *corrective* (correcting broken links where they are discovered) or *adaptive* (never storing actual links, only instructions for making them as required) (Ashman, 2000).

From the user's point of view, it might initially seem that preventative solutions are ideal, since the irritation of broken or misdirected links will never happen. However, many of the preventative measures can only guarantee accurate links within a limited scope and changes outside that scope (such as an entire domain name change) can still result in broken links. Also, they can be functionally limited; for example, it may be impossible to guarantee link integrity in information that is not part of the same preventative scheme.

Corrective solutions tend to be more robust, as they assume breakage will occur and have procedures in place to correct links, where possible, or to otherwise deal with them. These procedures are sometimes computations that aim to discover the new location for the linked document. These often function as mass correction procedures, taking place at intervals, which detect broken links and attempt to correct them, discard them, or notify the user of the problem. From the everyday user's point of view, this is a reasonable form of solution, requiring little or no effort on their part, with breakages often not encountered by the user. However, it is still possible that the user will discover a broken link, especially if it is some time since the most recent correction.

Also, the corrective approach may discard unfixable links that the user has previously required. This leaves the user with the knowledge that a link that was once present is now gone and seemingly unrecoverable. Perhaps a more user-friendly solution to irretrievably broken links is the so-called "soft 404s" (Bar-Yossef, Broder, Kumar, & Tomkins, 2004)—when pages go missing, those pages are replaced by human-readable error messages, which essentially assume the identity of the missing page. They frequently offer the user the option of a search of the site, or perhaps redirect the user to a new location. It is estimated that 25% or more of all dead links are these soft 404s (Bar-Yossef et al., 2004).

The correction of links is very much still a research problem, with much effort expended into double guessing the original link creator's intentions in making the link. However, if the link was originally created by a computation, then the correction often amounts to nothing more than executing the computation again. If, for example, the link computation was to "link every instance of a person's name to his or her home page," and then the name is moved within various pages, or appears or disappears, the links can easily be reinstated to their correct positions.

The adaptive approach assumes that all data is subject to change, as do the corrective solutions. Instead of having procedures for correction in place, it guarantees correct links by never having links in storage. The instructions to create links are stored, and links are created as required from them. This dynamic approach was trialled in experimental systems that created links either on user demand (Davis, Knight, & Hall, 1994) or as a background process (Verbyla & Ashman, 1994).

The adaptive approach is now evident in every-day web usage; for example, scripts are used to create and serve web pages dynamically, and often result from a database query, where links are calculated and inserted at the time of serving. However, it cannot assist with links created and maintained by other users and neither can the preventative or corrective approaches.

In the end, the major obstacle to link integrity in the web is its anarchic nature. There is no central authority that can impose robust linking practice on the mass of users; indeed the failure of the error-tolerant naming solutions such as URNs to capture the imagination shows that it is not even possible to tempt users into good linking practices, let alone force them.[11] Even a direct plea from one of the joint creators of the web has been unable to achieve link robustness (Berners-Lee, 1998). As long as users continue to create links without troubling to maintain them, other users will eventually encounter them as broken links. There are, however, numerous solutions that when combined will help keep the problem at bay, so that users may experience almost 404-free browsing.

Personalizing Links

In the previous section, we saw the utility of computation of links for reducing the number of broken links a user encoun-

[11]As evidenced by the lack of recent activity on URNs at the World Wide Web Consortium's *Naming and Addressing* page at http://www.w3.org/Addressing/

ters. In this section, we will see how computed links, with other technologies, allow the user to personalize the links and types of links, regardless of the source, type, or ownership of the data.

All users at some stage want to create their own links or annotations, generally to record their own associations between data or to ease everyday information access. In fact, easing information access is an obvious but frequently overlooked purpose for links—links function as a form of "user pull" of information, hiding the information while still making its presence known, but providing that information on request with the minimum of user effort.

Users have different needs, and an author of web pages cannot anticipate all such requirements, let alone provide them. Even if all the potentially useful links were provided, not only would users disagree on the facility of the links, but the interface and performance of the web would suffer.

For example, not every user wants a dictionary link, which could give a basic definition of any word selected by the user. However nonnative speakers of a language could find such a link invaluable. Glossary links are essential to a reader not familiar with technical terms, but become intrusive to seasoned readers. In each case, the users want to be able to "switch on or off" links to reflect their own needs.

This is easy with current research technology. Even as far back as 1992 (Davis, Hall, Heath, Hill, & Wilkins, 1992), the technology to provide links over data not owned by the user was available. It was even quite simple to create a form of computed link, for example, to link every occurrence of a name with a bibliography. Within a short time, it even became feasible not just to exploit link computation to automatically create one's own links, but to even create one's own link-computation specifications, so that exactly the right computation, pointing into the right data collections, could be easily linked at the user's wish (Verbyla et al., 1994).

The technology that supports personalization of links has not yet propagated into mainstream web browsers. Yet the different solutions have been trialled in a web context; for example, the Distributed Link Service enabled individuals to make their own private link sets or to contribute to their group's collective link sets (Carr, De Roure, Hall, & Hill, 1995). Even the creation of one's own link-computation specifications was trialled in a web environment (Cawley, Ashman, Chase, Dalwood, Davis, & Verbyla, 1995). Technically, it should be a simple matter to incorporate this feature into all browsers. Whether this happens depends largely on whether users insist on it.

Surprisingly, even the growing awareness of the need for privacy in online actions has not yet motivated users to demand an easy-to-use personalized link facility. Even personalized annotations are a minority facility, only feasible for the reasonably technically literate user. Being able to record one's personalized links enables users to record their private associations and collections, often representing original intellectual ideas or commercially valuable information without advertising it to all. As more of each user's online interactions and transactions are exposed to outside scrutiny, authorized or otherwise, the provision of a feature with the fortunate side effect of offering private means for recording ideas ought to become a priority for future web browser development.

SEARCHING AND QUERYING

Browsing, enabled by hypermedia links, was the first technology for accessing information on the web. However, the enormous success of the web and its rapid uptake by millions of users rapidly rendered browsing alone a manifestly inadequate tool for information access. Searching has now become a major, perhaps *the* major, means of locating information on the web.

However, accessing information via search engines presents difficulties. First, users face the problem of creating an accurate description of their requirement. Second, users must make sense of the results that the search engine produces. Solutions to these difficulties are discussed in this section.

Query Formulation

Experts agree that the typical user will experience difficulties when attempting to formulate an effective query. Three significant factors stand out:

1. *Low user commitment*. Users are reluctant to provide information beyond the bare minimum. In fact, what emerges is that search-engine use is characterized by short queries and limited interaction coupled with unreasonably high expectations (Jansen, Spink, & Saracevic, 2000; Rolker & Kramer, 1999; Kobayashi & Takeda, 2000):

   > Real people in the real world, doing real information seeking and in a hurry, use web search engines and give two-word queries to be run against billions of web pages. We expect, and get, sub-second response time and we complain when there are no relevant web pages in the top 10 presented to us (Browne & Smeaton, 2004).

Users, it would appear, are wary of commitment. They expect very high-quality results from every search instance rather than subscribing to the notion of search as an iterative process.

2. *Uncertain information needs*. Users often have an incomplete understanding of their information need, and their initial need will frequently mature during, and in direct response to, the process of searching (Lancaster, 1968):

   > Searchers normally start out with an unrefined or vague information need which becomes more sharply focused as their search continues and exposure to information changes their information need. (Browne et al., 2004).

This initial lack of clarity with regard to the object of the search readily translates into imprecise query terms, which in turn begets irrelevant and disappointing results. Some users quit the searching process at a very early stage, confounded by this apparent "failure."

3. *Difficulties in expression*. Users may not know the correct syntax to frame their query or the commands to interact with the search engine. They *may* know in general terms what

web pages they wish to retrieve, but struggle to find the query terms most likely to identify them:

> Except in special circumstances, it is difficult for a user to ask an information retrieval system for what they want, because the user does not, in general, know what is available and does not know from what it has to be differentiated. (Card, Robertson, & Mackinlay, 1991)

This is not surprising, as the formulation of a "successful" query requires some awareness of information-retrieval theory, relatively uncommon among typical users. Some of this theory can be guessed by observant users (who might notice, for instance, that query terms that frequently occur in web pages produce poor results), but the remaining users must persevere in an unfamiliar environment, never knowing why their search failed, or indeed why it succeeded.

Relevance Feedback

To counter some of the problems just mentioned, researchers working with search engines have developed strategies designed to improve the quality of a submitted query. One of these strategies is *relevance feedback* (RF):

1. The user submits an initial query and the search engine serves results.
2. The user then identifies *relevant* and *nonrelevant* web pages using associated checkboxes, clickable links, radio buttons etc. This action supplies the search engine with *feedback.*
3. The search engine then automatically *modifies* the original query in response to the feedback. This may involve adding search terms to the query, known as *query expansion*. It may also involve *reweighting* of the query, where information in the relevant and nonrelevant set of documents is used to modify the importance of various query terms.
4. The modified query is run by the search engine and a new set of search results is shown to the user.
5. This process continues until the user's information need is satisfied.

This technique is particularly successful, and it has been repeatedly demonstrated that relevance feedback can improve the performance of a search engine at comparatively little cost to the user or the system (Ruthven & Lalmas, 2003; Harman, 1992b; although see Spink, Jansen, & Ozmultu, 2000, for contrast). For example, Koenemann (1996) noted that the:

> . . . availability and use of relevance feedback increased retrieval effectiveness; and increased opportunity for user interaction with and control of relevance feedback made the interactions more efficient and usable while maintaining or increasing effectiveness.

However, it has also been found that in order for relevance feedback to be effective, the user *must* be offered a meaningful dialogue for participation. Ruthven recently conducted a series of RF experiments examining the performance of user-supplied feedback measured against an automated equivalent. Commenting on the relatively low performance of the users, he identified a failure in the infrastructure supporting the user's search:

> . . . simple term presentation interfaces are not sufficient in providing sufficient support and context to allow good query expansion decisions. Interfaces must support the identification of relationships between relevant material and suggested expansion terms and should support the development of good expansion strategies by the searcher.' (Ruthven, 2003)

In summary, many search-engine users find the formulation of an effective query difficult. Relevance feedback offers an effective solution, providing that:

- the users are willing to commit to several search iterations
- a suitable and useable feedback interface is implemented

The Results List

It is well accepted that the results generated by a web search engine will frequently not satisfy the user. Sometimes this negative result will occur because the search engine cannot find *any* documents that match the user's information need. However, it is much more likely to happen because the user has submitted a very broad query, resulting in too many documents. This common difficulty has been identified by one author as the "abundance problem," occurring when "the number of pages that could reasonably be returned as relevant is far too large for a human user to digest" (Kleinberg, 1999).

Many solutions to this challenge have been suggested, but three interesting approaches are (a) community-based ranking algorithms, (b) improved visual interfaces, and (c) document clustering. We now examine each approach in turn.

Community-Based Ranking Algorithms

A search engine normally ranks a set of web pages in order of the likelihood that they will be relevant to the user's information need, the document most likely to be relevant appearing first. This likelihood of relevance is usually calculated using a statistical measure related to the occurrence of the query terms in the documents concerned. Known as "term frequency," this measurement is traditionally normalized with respect to document length and multiplied by a measure reflecting the specificity of each term within the document collection (Sparck Jones, 1972; Aizawa, 2003; Salton, Yang, & Yu, 1975).

Document rankings dependent upon term frequency represent a purely arithmetical evaluation of the web pages concerned. This evaluation can (and does) provide a useful approximation of the likelihood of relevance to the user, but is by no means an authoritative measure. The underlying assumption-that term frequency translates directly into relevance—is exactly that: an assumption. There is no guarantee that a web page containing many query terms will be any more relevant than a second page containing a lower number of the same terms. The latter may have fewer of the important keywords, but might be far superior in other, less-quantifiable respects

(i.e., it may be more concise, better illustrated, superior in style, easier to read, more complete in its references, etc.).

One solution to this problem has been to supplement the rudimentary rankings that can be constructed through statistical observations with more sophisticated sources of information. This has led to the development of a general class of ranking algorithms that implement *citation based metrics* for relevance scoring (Garfield, 1972; Pinski & Narin, 1976). In these algorithms, the relative importance of each web page is a function of the number of other web pages that link to it (Brin & Page, 1998). As Kleinberg observed (1999), these "in-links . . . encode a considerable amount of latent human judgment." Accordingly, a web page that is referenced by a good proportion of its peers is a natural candidate for high ranking.

Improved Visual Interfaces

Hearst observed that "long lists littered with unwanted irrelevant material" represent an unwieldy and nonintuitive method for delivering search results to the user (Hearst, 1997). Rather than expending resources developing more sophisticated ranking algorithms, Hearst asserted that the answer to this problem lay in shifting:

> . . . the user's mental load from slower, thought intensive processes such as reading to faster, perceptual processes such as pattern recognition. It is easier, for example, to compare bars in a graph than numbers in a list. Color is very useful for helping people quickly select one particular word or object from a sea of others.

The author went on to speculate that in the near future advanced search interfaces may abandon the two-dimensional (2D) page metaphor altogether, adopting "alternatives that allow users to see information on the web from several perspectives simultaneously." A software implementation based on this very idea, known as the Information Visualizer (Card, Robertson, & Mackinlay, 1991) has proved surprisingly powerful; in one experiment an organizational hierarchy requiring 80 printed pages was displayed on just one three-dimensional (3D) screen (Robertson, Mackinlay, & Card, 1991). It seems clear that a visual tool for searching the web that helps the user to "see" a set of search results rather than just "read" them would have considerable utility.

Document Clustering

Another way of simplifying a set of search results is to generalize. The process of generalizing a set of results begins with identifying particular commonalities shared by some members of that set (i.e., use or nonuse of certain terms, subject matter, file type, etc.). web pages determined to be "similar" in some way are then grouped together under a single category or heading, and the user is subsequently presented with several coherent clusters of web pages rather than the traditional list. Provided the number of clusters is relatively low, this technique quickly reduces the cognitive load of studying the results, allowing a user to "skim" rather than read.

In a good example of this technique, Truran, Goulding, and Ashman (2005) described an attempt to solve the problem of ambiguity in query-based search. Lexical ambiguity causes considerable problems, complicating the already difficult task of establishing what a user is actually looking for. An analysis of the query log of a very large online encyclopedia recently found that 1 in 20 queries was largely wasted simply, because a user chooses ambiguous query terms (Wen, Nie, & Zhang, 2001).

Discussing this problem, the authors developed a sense categorization service known as SENSAI. This service was remarkable in that it was capable of auto-categorizing a set of search results into believable sense categories without a semantic analysis of the web pages concerned. Neither was an external knowledge source, such as a dictionary or an expert system, consulted. Instead, sense discrimination was the quiet background product of user consensus; the system studied and learned from its clientele, utilizing what Fitzpatrick et al. termed the "information-consuming habits" of its users (Fitzpatrick & Dent, 1997) to create a human-intuitive overview of the distinguishable semantics of the query, like documents grouped with like (Michalski, Stepp, & Diday, 1983).

Summary

Users of a search engine are frequently overawed by the number of results returned when a query is submitted, and rarely proceed beyond the first page of suggestions (Harman, 1992). To ensure that the most relevant documents appear on the first page, more sophisticated ranking algorithms have been developed, which exploit the inherently self-referential nature of the web.

However, even with these improvements, search engines are still serving large chunks of unrefined information to their users. This seems likely to change soon. Search engines will almost certainly begin to abstract categories and significant groupings from any given set of results automatically, and visual interfaces that simplify the user's task seem probable.

PERSONALIZATION, PORTALS AND COMMUNITIES

In most circumstances, one of the major impediments to effective use of the web is the disorientation that results from its sheer size. Statistics are commonly quoted as to just how vast the web is with one recent estimate being 11.5 billion pages (Gulli et al., 2005). This, no doubt, will become rapidly out of date, but the actual figures are not important from a user's perspective.

From the perspective of end users (i.e., content consumers), it is increasingly difficult to find what they are looking for amid the sea of inappropriate material. From the perspective of authors (i.e., content providers), it is becoming increasingly difficult to prevent their pages from being lost in the crowd. In this section we will examine approaches of personalization, portals, and community support that are becoming widely used to address these issues.

Although personalization, portals, and communities are quite distinct approaches to web design, they are all aimed at producing an end-user experience that is uniquely appropriate for each individual. This is what might be called "my web." *Personalization* is where the contents of websites are changed (either transparently or under user control), so that the material is appropriate for the needs and requirements of the individual. This involves a wide range of techniques of varying degrees of sophistication, from a user-selectable color scheme at its simplest through to an adaptive content-management system based upon a comprehensive user model. Such systems are commonly used for e-commerce websites, and have been used experimentally for e-learning. *Portals* are gateways to the web that aim to make its vastness more accessible by providing an appropriate set of links and default paths (usually they are continually or at least regularly updated). These may be personalized at the level of individuals, an organization, a special interest, or some combination of these. Where the personalization is operating at the level of organizations, special interests, or other groups, there is an opportunity for the personalized experience to be shared by a *community*. A major change in web design in recent years has been that there are now a variety of tools and techniques to both support community interaction on the web and leverage the power of collaborative effort.

Personalization

Until quite recently, the vast majority of web content has been designed for the needs of content providers rather than end users. In terms of a communication model, a traditional static HTML site has a lot in common with a poster on a billboard. It offers the same message to all passersby, regardless of whether or not it is appropriate to them. The only reason that this model is effective, on either a billboard or the web, is that there are (or at least there can be) many passersby. Modern websites frequently consist of a lot more than static HTML, they are often dynamic front-ends for a wide range of resources and services. However, their communication model is usually declarative—with a "one-size-fits-all" approach to content that treats the user as a hypothetical "average."

This state of affairs has recently begun to change, with the advent of websites that personalize content for each end user. One of the best known examples of this is Amazon.com, which stores information about the customers' interests—gleaned from various sources—to generate a personalized home page and suggest items that are likely to be of interest. Adaptive hypermedia (AH) (Brusilovsky, 2001) is an academic discipline that is dedicated to bringing personalization to the web. Most applications of AH techniques are either simple e-commerce applications, such as that just described or are in e-learning. Educational adaptive systems (AEH) (Brusilovsky, 2001b) have arisen as a logical progression of using the web for teaching and learning. In traditional education, the experience is very adaptive. A student who doesn't understand a question will ask the teacher for an explanation. If the student doesn't understand the answer, then the teacher explains it in different terms—maybe by simplifying, using an example, or drawing a picture. Exactly how the teacher explains it depends upon the needs of the student, so that the content is adapted for the individual student. Static web pages generally give the same experience to all students regardless of how effective this may be, good human teachers provide a customised experience for individuals. The ultimate goal of AEH systems is to provide this capability to web-based learning. So far, most such systems are experimental, but they are important because the majority of AH research has been done in this area. See Jameson (in this book) for more detail on adaptive interfaces.

Adaptive and Adaptable Systems

Two quite distinct approaches to the personalization of hypermedia are (Patel & Kinshuk, 1997; Wadge & schraefel, 2001) the adaptive and the adaptable. Adaptive systems dynamically organize their contents to meet the perceived needs of the user. This process is transparent, and does not require any direct intervention by the user. Adaptable systems make changes to the content (or more commonly the user interface) only as a result of the explicit intervention of the user.

Adaptable systems gather information explicitly from the user, such as iWeaver (Wolf, 2002). This is typically either in the form of some sort of "control-panel" setting, or a questionnaire (commonly completed as part of a user registration process) to elicit a set of preferences from the user. These preferences may be used to determine various aspects of the user interface (i.e., color schemes, screen layout etc), or they may be about the background or perceived needs of the user (i.e., knowledge or areas of interest).

Adaptive systems gather information implicitly by monitoring the behavior of the user (Moore, Stewart, Zakaria, & Brailsford, 2003). The navigation of the user through the system may be monitored (i.e., the links followed) or a quiz may be used may be used to attempt to deduce information about users (i.e., their abilities). The distinction between this and an adaptable system is that the adaptable system is asking users what they want; whereas the adaptive system attempts to deduce what the user needs (not necessarily the same thing) and provide it transparently. Adaptive hypertext is really a type of "attentive system" (Maglio, Barrett, Campbell, & Selker, 2000), which is a system that pays attention to the user's behavior, so that the user can respond appropriately. Because there is always a risk of user behavior being misinterpreted by the system, attentive systems should not be used for critical information (Maglio et al., 2000). The adaptation may well improve the usability and appropriateness of a website, but it is important that any essential information is available independently of any adaptation.

Although these two approaches to personalization are quite distinct, it is possible for a system to be both adaptive and adaptable. Amazon.com is a good example of this because it acquires information from users both implicitly and explicitly. It uses information based upon purchases and follow ups of search results or suggestions (i.e., click-through), all of which is implicitly derived from user behavior (i.e., from this point of view it is adaptive). However, it also allows users to explicitly rate items, and it uses these ratings as a basis for future suggestions (i.e., from this point of view it is adaptable). AEH systems also can use both methods, for example WHURLE (Brailsford, Ashman, Stewart, Zakaria, & Moore, 2002).

Need or Desire?

Adaptive hypertext systems base their content delivery upon what the system perceives to be a user's need. Adaptable systems, on the other had, use information obtained directly from the user, which in most case will represent that user's desire. Since need and desire are not necessarily the same thing, there is potential for a conflict of interests. When considering adaptive systems, the underlying assumption is that the system knows better than the user what is most appropriate. Although this sounds disempowering (and if implemented badly it can be), there are many circumstances where this view is justified, particularly if the user doesn't understand the ramifications of a particular choice at a decision point. For example, in educational systems learners may not always have all of the facts or the pedagogic background necessary to take control of their own teaching regime. In "real-life" education, this is why we have teachers rather than just resources. Consider, for instance, a student who has problems with the math in a subject area that makes extensive use of math, although it is not the primary topic of study (engineering or computer science would be good examples here). Given the choice, many such students would prefer fewer math problems; however, what they probably need are *more* (and better explained) math problems. An adaptive system that "knows best" would thus be capable of transparently providing remedial exercises (as would a real-life teacher who spotted the problem). However, an adaptable system runs the risk that the student might choose to avoid the parts that they find the most onerous.

Implementing this "system-knows-best" paradigm without due care creates the potential for serious problems in an adaptive system. It is quite likely to result in frustration or anger if users realize that the system is going against their desires or if the system's assumption of need is incorrect—particularly a problem with a poorly thought-out or incomplete user model (see "User Modeling"). Well-designed adaptive systems do not take control away from the user; rather they provide suggestions (albeit sometimes strong ones) as to the most appropriate path. This is then a user interface issue, and it may take the form of reordering content or suggesting what content is considered most appropriate (i.e., a common mechanism is to change link color).

Methods and Models of Adaptation

There are two main approaches to implementing adaptation on the web, at the level of either content (i.e., adaptive presentation) or linking (i.e., adaptive navigation support). Within each of these, there are a number of techniques used that are described briefly here but are reviewed in detail elsewhere (Brusilovsky, 1996).

The majority of adaptive presentation operates on text, although in principle any media type may be adapted. Adaptation of modality (Brusilovsky, 1996) is where the adaptation operates at the level of choices between different media types (i.e., text, speech, animation etc.). Although this will likely become important in the future, multimedia adaptation is currently very rarely implemented (one of the few examples is Fagerjord, 2005). There are a number of ways in which the adaptive presentation of text has been implemented, usually by manipulating fragments of predefined text. Some systems add fragments to a standard minimum body of text; others remove fragments that the system deems inappropriate. When fragments are removed they may be completely hidden, or they may be merely dimmed. Another variant of fragment removal is to use "stretchtext" (a body of text is hidden behind a single word or phrase and expanded upon request). Instead of adding or removing fragments of text, some systems simply reorder the fragments. A major problem with all forms of adaptive-text presentation is the impact that it has upon narrative flow. Different users will necessarily have different narratives, which isn't a problem *per se* other than the fact that it is possible (or even likely) that a web page will no longer form a coherent whole if its text is manipulated. This problem may be ameliorated by writing text in the form of conceptually discrete "atoms" (Brailsford, Stewart, Zakaria, & Moore, 2002), but that causes quite serious problems with authoring (most people find it difficult to write following this sort of constraint) and makes legacy content difficult to adapt.

Adaptive navigation support has fewer problems with narrative flow, because links are (usually) less fundamental to the sense of a page than is text. Adaptive linking has been divided into five main categories (Brusilovsky, 1996): direct guidance, link sorting, link hiding, link annotation, and map adaptation. With *direct guidance*, the system makes a recommendation as to the next node to visit, based upon the adaptation criteria. This is often used for a "guided-tour" type of direction for novice users (i.e., it is tantamount to a linear sequence of "Next" buttons). With *link sorting*, the ordering of links is changed so that the most appropriate one is at the head of the list or in a particularly prominent place. Inappropriate links are not displayed in link hiding. This, of course, can make parts of the system inaccessible to some users—which may or may not be acceptable. A variant of link hiding is *link dimming*, where it is still possible to follow the link, but it is less prominent in the user interface than more relevant links. In a link-annotation system, visual clues are attached to links to give users some idea of their status with respect to the system; for example, font sizes or colors can be used to provide a gradation of relevance. Lastly, in map-adaptation systems a navigational map is generated according to the adaptation rules.

User Modeling

In order to be able to do any adaptation, adaptive systems need to know who is using them and store information about that person. This information is stored in a "user profile" either on the local machine or, far more commonly, in a centralized database. In the user profile, as well as personal information (name, ID etc.), the criteria that the system will use for adaptation are stored and constitute the user model. Although a detailed discussion of user modelling is beyond the scope of this chapter, the basic principles are very simple. A user model is simply a set of rules that adaptive systems use to make decisions about relevance. One approach to user modelling is that of *stereotype* models (Kaplan, Fenwick, & Chen, 1993). With these,

users are grouped together with other like-minded individuals. For example, "second-year chemistry undergraduates" or "people who like cats" are both stereotypes and, as such, it is possible to make reasonable assumptions about members of these groups (i.e., "will know the structure of butane" or "will appreciate photos of cute kittens" respectively). The other main user-modeling methodology is the *overlay* model (Fischer, Mastaglio, Reeves, & Rieman, 1990), where the individual's knowledge, background, and any other relevant data are overlaid onto the sum of all knowledge in the system. Usually, adaptive-hypermedia systems adopt one of these two models. However, they do not have to be mutually exclusive, as it is possible to combine the two approaches so that any given user is initially stereotyped. Then, as the system learns more about them, it uses an overlay model to record their progress (Zakaria & Brailsford, 2002).

User modeling is critical for a usable adaptive system. If the user model is in some way flawed (i.e., if the information stored does not match the reality of users' background and behavior), then content is likely to be maladapted for a user and the system is often extremely difficult to comprehend. This is compounded by the fact that there are no standards for user modelling, and most adaptive systems use their own proprietary user model.

Authoring

To date there are few, if any, mature authoring systems for personalized web systems (AHA! (De Bra, Stash, & Smits, 2004) is one of the more mature authoring systems). This is, no doubt, because most adaptive systems are either experimental or proprietary, and in both cases the authors are either members of, or have direct access to, the development team. However, there are a number of potentially difficult issues that need to be considered when attempting to design authoring interfaces for adaptive web-based systems (Cristea, Carro, & Garzotto, 2005). First, there is a need for a new design paradigm, together with associated metaphors because personalization breaks the fundamental design paradigm of the web, that of the "page." When authoring a conventional web page, the author concentrates on content, and can reasonably assume that users will see the page as it is created. Hence, most authoring systems for static web pages use a conventional WYSIWYG paradigm, which is quite similar to that of a word processor. In the case of adaptive systems, however, the authoring process requires a separation of content authoring from the authoring of the parameters for adaptation. Depending upon the adaptation model that is used, what the author sees might be somewhat or completely different from what any given end user sees.

Recently, research has started to address these issues, and frameworks (such as LAOS, Cristea, 2005) have been developed for adaptive educational systems that allow the separation of different aspects of the authoring process. The LAOS framework has five layers, which result in different authors being able to create different sections or aspects of adaptive content. For example, there is a "goals & constraints" model where a pedagogic expert can use the content from the "domain model" to structure a lesson that makes use of a "user model," "adaptation model," and a "presentation model." This layered structure not only decreases individual author load, but also encourages the reuse of existing materials (whether they are domain content, adaptation rules, or presentation settings). The great benefit of this approach is that it can combine content created by different authors working at different levels. However, that is also its weakness in that care must be taken to avoid inappropriate use of content originally intended for a different context. Adaptive authoring systems are very much in their infancy, but although there are no mature systems, the current research is at least starting to discover the issues and problems involved with their design.

Portals

A common way to gain a personalized experience of the web is by the use of portals. These are gateways to online resources that are, at their most basic, simply a set of links that provide starting point for web use. Usually, though, they are highly sophisticated systems that are continually updated and highly personalizable. Portals are hugely popular, at the time of writing of the top 10 most browsed English language websites, four are portals,[12] and they are often presented as a way for users to create "their space" on the web. They are usually adaptable, allowing for explicit selections of user interface options or content appropriate for specific hardware platforms (i.e., low graphics content for PDAs), although some portals also use techniques of adaptivity (*q.v.*).

Many portals have their roots in either Internet directories or search engines, and therefore a common model of a portal user interface is that of a categorized front page (i.e., a "list of lists"). Some portals are essentially starting points for a collection of resources—typically these are intranet systems where the facilities determined for a particular user are presented on login (i.e., different staff or student resources in a university system). Some portals use semantic web technology to categorize their contents (i.e., Ontoportal[13]).

Communities

The Internet has its roots in communities, in creating links and bridging geographic, cultural, or temporal barriers. Communities on the web are often supported by, or generated through, portals, which provide starting points for people of similar interests. Not only do these provide channels through which users can navigate to find resources, but in doing so they can bring about a shared cultural space. Portals, especially those that are intranet based, are often designed to support communities.

Part of the original vision of computer pioneers, such as Nelson (1982) was that people could easily publish, edit, and comment on as well as read publicly available materials, and this is

[12]Alexa Web Search: top sites. http://www.alexa.com/site/ds/top_sites

[13]Ontoportal—http://www.ontoportal.org.uk/

now starting to become a reality on the modern web. Weblogs (more commonly referred to as blogs) allow a user to post material, which can then be cited, commented upon, and redistributed throughout the internet. These blogs require no special knowledge of authoring and are usually accessed with a standard web browser.

Another technology that is important in the development of communities on the web are Wikis,[14] which are community websites designed to facilitate collaborative authoring. The most spectacular example of this is Wikipedia, a vast encyclopaedia written collaboratively by volunteers using a Wiki.[15] This project was founded in 2001, and in 2004 it contained over a million articles in more than 100 languages. The system provides facilities to discuss changes made to articles, and authors may vote on controversial issues, the editorial assumption being that the exposure of material to many users will result in the addition of depth and elimination of errors.

CONCLUSIONS

In this chapter we have considered HCI in a specific web context, looking at the issues impeding smooth user interaction with web tools and documents. While some problems remain unsolved, generally the usability of the web has improved greatly since its early days. While many challenges remain, there are at least clear research directions indicating potential solutions.

References

Abascal, J., & Nicolle, C. A. (Eds.). (2001). *Inclusive design guidelines for HCI*. Taylor & Francis.

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management, 39*(1), 45–65.

Ashman, H. (2000). Electronic document addressing—dealing with change. *Computing Surveys, 32*(3), 201–212.

Bar-Yossef, Z., Broder, A. Z., Kumar, R., & Tomkins, A. (2004). Sic transit gloria telae: Towards an understanding of the web's decay. In *Proceedings of the 13th International Conference on World Wide Web* (New York, NY, USA, May 17–20, 2004) WWW '04 (pp. 328–337). New York, NY: ACM Press.

Bergman, M. K. (2001). The deep web: Surfacing hidden value. *The Journal of Electronic Publishing*. Retrieved April 23, 2007, from http://www.press.umich.edu/jep/07-01/bergman html

Berners-Lee, T. (1998). Cool URIs Don't Change, http://www.w3.org/Provider/Style/URI.

Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V., & Oinas-Kukkonen, H. (1997). Fourth generation hypermedia: Some missing links for the World Wide Web [Special issue on HCI and the web]. *International Journal of Human-Computer Studies, 47*(1), 31–65.

Brailsford. D. (1999). Separable hyperstructure and delayed link binding. *ACM Computing Surveys, 31*.

Brailsford, T. J., Ashman, H. L., Stewart, C. D., Zakaria, M. R., & Moore, A. (2002). *User control of adaptation in an automated web-based learning environment*. First International Conference on Information Technology & Applications (ICITA 2002), Australia.

Brailsford, T., Stewart, C., Zakaria, M., & Moore, A. (2002). Autonavigation, links and narrative in an adaptive web-based integrated learning environment. *Eleventh International World Wide Web Conference*.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Proceedings of the Seventh International Conference on World Wide Web* (*WWW7*), 107–117.

Browne, P., & Smeaton, A. (2004). Video information retrieval using objects and ostensive relevance feedback. In *Proceedings of the 2004 ACM Symposium on Applied Computing* (pp. 1084–1090). ACM.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modelling and User-Adapted Interaction, 6*, 87–129.

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modelling and User Adapted Interaction, 11*(1/2), 87–110.

Brusilovsky, P. (2001). Adaptive educational hypermedia (invited talk). The Tenth International PEG *Conference*, 8–12.

Burnett, G. E., Summerskill, S. J., & Porter, J. M. (2004). "On-the-move" destination entry for vehicle navigation systems—unsafe by any means? *Behaviour and Information Technology, 23*(4), 265–272.

Cailliau, R., & Ashman, H., (1999). A history of hypertext in the web. *Computing Surveys Special Issue on Hypertext and Hypermedia, 31*, ACM.

Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (*CHI '91*) (pp. 181–186). ACM.

Carr, L., De Roure, D., Hall, W., & Hill, G. (1995). The distributed link service, a tool for publishers, authors and readers. *Paper presented at the 4th International WWW Conference*. Retrieved April 23, 2007 from http://www.w3.org/Conferences/WWW4/Papers/178/

Cawley, T., Ashman, H., Chase, G., Dalwood, M., Davis, S., & Verbyla, J. (1995). A link server for integrating the web with third-party applications. *Proceedings of the First Australasian World Wide Web Conference*, Australia. Retrieved, from http://ausweb.scu.edu.au/aw95/integrating/cawley/index.html

Cristea, A. (2005). Authoring of adaptive hypermedia: Adaptive hypermedia and learning environments. In S. Chen & G. Magoulas (Eds.), *Advances in web-based education: Personalized learning environments* (pp. •••). IDEA Publishing Group.

Cristea, A., Carro, R., & Garzotto, F. (2005). A3EH: The 3rd workshop of Authoring of Adaptive and Adaptable Educational Hypermedia, (http://wwwis.win.tue.nl/~acristea/AAAEH05/papers/editorial.pdf)

Davis, H.C. (1995). To embed or not to embed. *Comms of the ACM, 38*(8), 108–109.

Davis, H. C. (1998). Referential integrity of links in open hypermedia systems. In *Proceedings of Hypertext '98* (pp. 207–217). ACM.

Davis, H., Hall, W., Heath, I., Hill, G., & Wilkins, R. (1992). Towards an integrated information environment with open hypermedia

[14]Wiki Wiki Web—http://c2.com/cgi/wiki?WikiWikiWeb

[15]Wikipedia—http://www.wikipedia.org/

systems. In *Proceedings of the Second European Conference on Hypertext* (pp. 181–190). ACM.

Davis, H., Knight, S., & Hall, W. (1994). Light hypermedia link services: A study of third party application integration. In *Proceedings of ECHT '94* (pp. 41–50). ACM.

De Bra, P., Stash, N., & Smits, D. (2004, August 23). Creating adaptive applications with AHA!, Tutorial for AHA! version 3.0. Tutorial at the AH2004 Conference, Eindhoven. Computing Science Report 04-20, 2004.

Eason, K. D. (1984). Towards the experimental study of usability: Ergonomics of the user interface. *Behaviour and Information. Technology, 3*(2), 133–143.

Fagerjord, A. (2005). Editing stretchfilm. *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia* (p. 301). ACM.

Fischer, G., Mastaglio, T., Reeves, B., & Rieman, J. (1990). Minimalist explanations in knowledge-based systems. *Proceedings of the 23rd Annual Hawaii International Conference on System Sciences*, 309–317.

Fitzpatrick, L., & Dent, M. (1997) Automatic feedback using past queries: Social searching? In *Proceedngs of the 20th Annual International Conference on Research and Development in Information Retrieval* (pp. 306–313). ACM.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science, 178*, 471–479.

Gulli, A., & Signorini, A. (2005). The indexable web is more than 11.5 billion pages. Poster at 14th International WWW Conference (WWW2005).

Harman, D. (1992). Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1–10). ACM.

Harman, D. (1992). Relevance feedback and other query modification techniques. In W. B. Frakes & R. Baeza-Yates (Eds.), *Information Retrieval. Data Structures & Algorithms* (pp. 241–263). Prentice-Hall.

Hearst, M. (1997). Interfaces for searching the web. *Scientific American, 276*(3), 68–72 (5).

International Organisation for Standardisation (1998). ISO 9241 Ergonomics requirements for work with visual display terminals (VDTs)—Part 11 Guidance on Usability.

Jansen, B. J., Spink, A., & Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing and Management,.36*(2), 207–227.

Jul, S., & Furnas, G. W. (1997). Navigation in electronic worlds: A CHI 97 workshop. *SIGCHI Bulletins, 29*(4).

Kaplan, C., Fenwick, J., & Chen, J. (1993). *Adaptive hypertext navigation based on user goals and context*. User Modeling and User-Adapted Interaction. Kluwer Academic.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46*(5), 604–632.

Kobayashi, M., & Takeda, K. (2000). Information retrieval on the web. *Computing Surveys* , 32(2), 144–173.

Koenemann, J. (1996). Supporting interactive information retrieval through relevance feedback. In *Conference companion on human factors in computing systems* (pp. 49–50). ACM.

Lai, J., Cheng, K., Green, P., & Tsimhoni, O. (2002). On the road and on the web? Comprehension of synthetic and human speech while driving. In *Proceedings of SIGCHI 2002*. ACM.

Lawrence, S., Pennock, D. M., Flake, G. W., Krovetz, R., Coetzee, F. M., Glover, E., et al. (2001). Persistence of web references in scientific research. *Computer*, *IEEE, 34*(2), 26–31.

Lazar, J., Bessiere, K., Ceaparu, I., Robinson, J., & Shneiderman, B. (2003). Help! I'm lost: User frustration in web navigation. *IT & Society, 1*(3), 18–26.

Maglio, P., Barrett, R,. Campbell, C., & Selker, T. (2000). Suitor: An attentive information system. *Proceedings of the 5th International Conference on Intelligent User Interfaces*, 169–176.

McCracken, D. D., & Wolfe, R. J. (2004). *User-centred website development: A human-computer interaction approach*. NJ: Pearson Prentice Hall.

Michalski, R., Stepp, R., & Diday, E. (1983). Automatic construction of classifications: Conceptual clustering versus numerical taxonomy. *Transactions on Pattern Analysis and Machine Intelligence*, *IEEE*, *5*, 528–552.

Moore, A., Stewart, C. D., Zakaria, M. R., & Brailsford, T. J. (2003). WHURLE—an adaptive remote learning framework. *International Conference on Engineering Education* (*ICEE-2003*), Valencia, Spain.

Nelson, T. H. (1982). *Literary machines*. Eastgate Systems Inc.

Nielson, J. (2000). *Designing web usability*. New Riders Publishing.

Otter, M., & Johnson, H. (2000). Lost in hyperspace: metrics and mental models. *Interacting with Computers, 13*, 1–40.

Patel, A., & Kinshuk, (1997). Intelligent tutoring tools in a computer integrated learning environment for introductory numeric disciplines. *Innovations in Education and Training International Journal, 34*(3), 200–207.

Pinski, G., & Narin, F. (1976). Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing and Management, 12*, 297–312.

Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (*CHI '91*) (pp. 189–194). ACM.

Ruthven, I. (2003). Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 213–220). ACM.

Ruthven, I., & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review, 18*(2), 95–145.

Salton, G., Yang, C. S., & Yu, C. T.(1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science, 26*(1), 33–44.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–20.

Spink, A., Jansen, B., & Ozmultu, H. (2000). Use of query reformulation and relevance feedback by Excite users. *Internet Research: Electronic Networking Applications and Policy, 10*(4), 317–328.

Truran, M., Goulding, J., & Ashman, H. (2005). Co-active intelligence for information retrieval. In *Proceedings of ACM Multimedia '05* (pp. 547–550). ACM.

Verbyla, J. L. M., & Ashman, H. (1994). A user-configurable hypermedia-based interface via the functional model of the link. *Hypermedia, 6*(3), 193–208.

Wadge, W. W., & schraefel, m.c. (2001). A complementary approach for adaptive and adaptable hypermedia: Intensional hypertext. In *Hypermedia: Openness, structural awareness, and adaptivity.* International Workshop OHS-7, SC-3, and AH-3, Aarhus, Denmark.

Wen, J., Nie, J., & Zhang, H. (2001). Clustering user queries of a search engine. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 162–168). ACM.

Wolf, C. (2002). iWeaver: Towards an interactive web-based adaptive learning environment to address individual learning styles. *European Journal of Distance Learning* (*EuroDL*). Retrieved April 23, 2007, from, http://www.eurodl.org/materials/contrib/2002/2HTML/iWeaver.htm

Zakaria, M. R., & Brailsford, T. J. (2002). User modelling and adaptive educational hypermedia frameworks for education. *New Review of Hypermedia and Multimedia, 8*, 83–97.

•29•

HUMAN-CENTERED DESIGN OF DECISION-SUPPORT SYSTEMS

Philip J. Smith
The Ohio State University

Norman D. Geddes
Applied Systems Intelligence, Incorporated

Roger Beatty
American Airlines

Introduction **574**
Approaches to Design **575**
Cognitive Task Analyses and Cognitive Walkthroughs 575
Work-Domain Analyses 577
Approaches to Design—Summary 577
Human Performance on Decision-Making Tasks **577**
Errors and Cognitive Biases 578
Slips 578
Mistakes 578
Cognitive Biases 578
Designer Error 579
Systems Approaches to Error 579
Errors and Cognitive Biases—Implications for Design 579
Human Expertise 580
Descriptive Models 580
Normative Optimal Models 581
Human Expertise—Implications for Design 581
Additional Cognitive Engineering Considerations . . . 582
The Human Operator as Monitor 582
Complacency and Overreliance 582
Excessive Mental Workload 582
Lack of Awareness or Understanding 582
Lack of Trust and User Acceptance 583
Active Biasing of the User's Cognitive Processes 583
Distributed Work and Alternative Roles 583
Organizational Failures 583
Case Studies **584**
Case Study A—Distributed Work in the National Airspace System: Ground Delay Programs 584
The Application Area 584
The Original Approach to Distributing Work in a GDP 584
An Alternative Approach to the Design of GDPs 584
Strategy for distributing work 584
Ration-by-schedule (RBS) 585
Slot swapping 586
Adapting to deal with uncertainty 587
Administrative controls 587
Evaluation of the Implemented Solution 588

Case Study A—Conclusions 588
Case Study B—Human Interactions with Autonomous Mobile Robotic Systems 589
The Application Area 589
The Need for Decision Aiding 590
The Design Solution 590
Results 592
Performance 592
Acceptance 592
Communication of beliefs 593
Communications of intentions in the context of a bid 593
Communication of direct commands 593
Case Study C—Interactive Critiquing as a Form of Decision Support 593
The Application Area 593
Diagnosis as a generic task 593
Sample problem 594
The need for decision aiding 595
The Design Solution 595
Critiquing 2 additional design considerations . . . 595
Complementary strategies to reduce susceptibility to brittleness 596
Evaluation of the Implemented Solution 596
Conclusion 596
References 597

INTRODUCTION

Computers can assist decision makers in a variety of different ways. They can, for instance, provide improved access to information or more informative displays of this information, or support more effective forms of communication. They can also use algorithms to actively monitor situations and to generate inferences in order to assist with tasks such as planning, diagnosis, and process control. This chapter focuses on interaction design issues (Preece, Rogers, & Sharp, 2002; Scott, Roth, & Deutsch, 2005) associated with this latter role, in which the software uses numerical computations and/or symbolic reasoning to serve as an active decision-support system or DSS (Turban, Aronson, & Liang, 2004).

The chapter begins with a focus on design methods, emphasizing the value of concrete scenarios and use cases (Bittner, 2002; Cockburn, 2000; Kulak & Guiney, 2003) to define the design problem, and the use of cognitive task analyses and work-domain analyses to guide both the development of the underlying functions in the DSS and the design of the interface that mediates the interactions of the user with these functions.

The chapter then proceeds to discuss aspects of human performance relevant to the design of DSSs. This includes a discussion of factors that limit human performance, as these factors suggest opportunities for improving performance through the introduction of a DSS. This discussion also includes consideration of the influence of a DSS on the user's cognitive processes, emphasizing the need to think in terms of the resultant joint cognitive system that is not a simple combination of the capabilities of the two agents (user and DSS) alone (Hoc, 2000; Hollnagel & Woods, 2005; Jones & Jacobs, 2000).

The discussion further emphasizes the need to consider more than just the surface interactions of the user with the interface to the DSS, and to consider the impact of the broader task context on performance (Amalberti, 1999; C. Miller, Pelican, & Goldman, 2000; Parasuraman, 2000; Rasmussen, Pejtersen, & & Goldstein, 1994). This latter emphasis suggests that the design of a DSS needs to be viewed from the perspective of cooperative problem solving, where the computer and the person interact with and influence each other (Beach & Connolly, 2005; Hoc, 2000; Jones & Mitchell, 1995; Larson & C. Hayes, 2005; Mital & Pennathur, 2004; Parasuraman, Sheridan, & Wickens, 2000; P. J. Smith, McCoy, & Layton, 1997). From this viewpoint, the underlying functionality is just as important as the surface-level representation or interface, as the overall interaction determines ultimate performance.

The chapter also frames the design and use of a DSS as a form of cooperative work between several individuals (Bowers, Salas, & Jentsch, 2006; Grabowski & Sanborn, 2003; Hutchins, 1990, 1995; Orasanu & Salas, 1991; G. M. Olson & J. S. Olson, 1997; Rasmussen, Brehner, & Leplat, 1991; P. J. Smith, Billings, et al., 2000), with the computer as the medium through which they cooperate. This teamwork may involve system users who are physically or temporally distributed. In addition, the discussion further suggests that it is useful to think of the design team as working cooperatively with the users, trying to communicate with them and extend or augment their capabilities, and doing so through the software and other artifacts (such as paper documents) that they have developed. This perspective is useful as a reminder that we need to consider the psychology of the designers as well as the psychology of the users of systems, applying that understanding to help us take advantage of the strengths of both groups as they work cooperatively.

Thus, this chapter approaches the design of DSSs from four perspectives. The first emphasizes the need to identify effective approaches to support the design process, including the application of use cases, cognitive task analyses, and work-domain analyses. The second is to provide a discussion of the human factors considerations relevant to the design of a DSS (Helander, Landauer, & Prabhu, 1997; Meister & Enderwick, 2002; Wickens, J. Lee, Liu, Becker, & Gordon, 2004). The third is to provide very practical lists of questions that need to be asked as part of the design process. The fourth is to use case studies to provide concrete, detailed illustrations of how these considerations can be addressed in specific application contexts.

APPROACHES TO DESIGN

A number of complementary and overlapping approaches have been articulated to help ensure adequate consideration of user needs, preferences, and capabilities during the design process, and to gain insights into possible functionality to incorporate in a DSS. These include the use of participatory design (Bodker, Kensing, & Simonsen, 2004; Schuler & Namioka, 1993), scenario-based design (Carroll, 1995), and the completion of various forms of needs assessments (Tobey, 2005; Witkin & Altshuld, 1995), cognitive task analyses (CTAs) and cognitive walkthroughs (Crandall, Klein, & Hoffman, 2006; Jonassen, Tessmer, & Hannum, 1999; Kirwan & Ainsworth, 1992; Rubin, Jones, & Mitchell, 1988; Schragen, Chipman, & Shalin, 2000). It also includes various forms of domain analyses aimed at identifying the critical features and constraints that characterize the domain of interest.

Previous chapters have discussed the use of participatory and scenario-based design as approaches to the design process. The focus of this section will therefore be on various forms of cognitive task and domain analyses. However, it should be noted that the development and results of a cognitive task analysis can in general be very useful in helping to support participatory design, as can the development and use of scenarios. As Mack (1995) noted: "Developing and sharing a set of touchstone stories is an important cohesive element in any social system. . . . Gathering, discussing, and sharing these stories as a group can be an effective means to team building" (p. 368).

Cognitive Task Analyses and Cognitive Walkthroughs

Scenarios and use cases are also important in providing the framing for specific cognitive task analyses, as descriptions or predictions of performance need to be considered in the context of specific tasks and task settings. The starting point for a specific cognitive task analysis is the high-level context provided by a scenario or use case, including the social setting, resources, and goals of users. Thus, to begin a cognitive task analysis, the starting point is specification of the person for whom the product is to be designed (a persona), what goal that person is trying to achieve, and the immediate and broader contexts in which that goal will be pursued.

A number of variations for conducting a cognitive task analysis (Gordon & Gill, 1997; Hollnagel, 2003) have their roots in hierarchical task analysis techniques, decomposing some high-level goal or task into a hierarchy of subgoals (Annett, Duncan, Stammers, & Gray, 1971; Annett & Stanton, 2000; Diaper & Stanton, 2004; G. A. Miller, Galanter, & Pribram, 1960; Preece, Rogers, Sharp, & Benyon, 1994). This approach, originally applied to pure task analyses that were not concerned with representing the underlying cognitive processes, has been extended in a number of ways that maintain a central role for describing goal-subgoal relationships, but that also introduce constructs or methods for describing associated cognitive processes.

One influential example of this evolution is the GOMS (Goals, Operators, Methods and Selection rules) model (Card, Moran, & Newell, 1983; Kieras, 1991, 2004; St. Amant, Freed, Ritter, & Schunn, 2005; Williams, 2005), which makes use of a hierarchical decomposition, but provides a control structure that introduces operators and rules for selecting which operator to apply in a given goal-driven context. As Preece et al. (1994) noted:

> A GOMS analysis of human-system interaction can be applied at various levels of abstraction in much the same way that the hierarchical task analysis splits tasks into subtasks. . . . Three broad levels of granularity determine the GOMS family of models:
>
> - The GOMS model, which describes the general methods for accomplishing a set of tasks.
> - The unit task level, which breaks users' tasks into unit tasks, and then estimates the time that it takes for the user to perform these.
> - The keystroke level, which describes and predicts the time it takes to perform a task by specifying the keystrokes needed (pp. 419–420).

This move toward explicit computational representation of internal processes has been extended to the point where broad, task-independent cognitive architectures have been developed that can be used to model performance in a given task/system context and generate performance predictions (Byrne & Kirlik, 2005). Three such systems are ACT-R (Anderson, 1993; Anderson et al., 2004), SOAR (Laird, Newell, & Rosenbloom, 1987; Nason, Laird, & Schunn, 2005), and EPIC (Kieras & Meyer, 1997; St. Amant et al., 2005). All three incorporate procedural knowledge that must operate within the constraints of certain constructs, such as working memory (Baddeley, 1998, 1999; Cary & Carlson, 2001; Radvansky, 2005; Seamon, 2005), that are used to model human capabilities and limitations (Meyers, 2004; Wickens et al., 2004).

This work on the development of computational models to describe or predict task performance has been complemented by more qualitative, judgment-based approaches using expert reviews that are similarly rooted in the hierarchical decomposition of a task based on goal-subgoal relationships. These latter techniques are motivated by the assumption that expert judgments about the adequacy of a design will be more accurate and complete if they are made while explicitly considering the context that the user will be dealing with (the user's current goal and subgoal and the display the user is viewing while trying to achieve that goal).

There are a number of variations on how to conduct such qualitative cognitive walkthroughs or predictive cognitive task analyses (Annett & Stanton, 2000; Bisantz, Roth, & Brickman, 2003; Diaper & Stanton, 2004; Gordon & Gill, 1997; Hollnagel, 2003; Jonassen et al., 1999; Kirwan & Ainsworth, 1992; Klein, 2000; Schraagen et al., 2000; Shepherd, 2000; Vicente, 1999). As an example, one such method was outlined by Lewis and Wharton (1997) as follows:

Step 1: Select the context for a use case to be used for an evaluation. Each such use case specifies three primary considerations:

- What are the characteristics of the user of the product that might affect its use? (the persona in the use case)
- What is the broader physical, organizational, social and legal context in which the product will be used?
- What is the high-level task or goal for which the product is being used?

Note that this specification is solution-independent. It essentially defines the design problem for this use case.

Step 2: Specify the normative (correct) paths for completing the goal of this use case (represented as a goal hierarchy) using the current product design, indicating the alternative sequences of steps that the user could take to *successfully* achieve the specified goal. Note that there could be more than one correct path for completing a given task.

Step 3: Identify the state of the product and the associated "world" at each node in the goal hierarchy. In the case of a software product, the state of the product would be the current appearance of the interface and any associated internal states (such as the queue of recently completed actions that would be used should an undo function be applied). The state of the "world" applies if the product or the user actually changes something in the world as part of an action, such as changing the temperature of a glass-manufacturing system through a process-control interface.

Step 4: Generate predictions. For each correct action (node in hierarchy), attempt to:

- Predict all the relevant success stories;
- Predict all the relevant failure stories;
- Record the reasons and assumptions made in generating these stories;
- Identify potential fixes to avoid or assist in recovery from failure stories.

This final step involves walking through the hierarchy along each path that leads to success, and, playing psychologist, generate predicted behaviors at each node (for each associated screen display or product state). This walkthrough could be done by an expert in human-computer interaction or by a domain expert. Because of the differences in their backgrounds relevant to the design and the domain of interest, individuals from each group are likely to generate somewhat different predictions.

Lewis and Wharton (1997) suggested asking four questions to help guide this prediction process:

- "Will the user be trying to achieve the right effect?"
- "Will the user notice that the correct action is available?"
- "Will the user associate the correct action with the desired effect?"
- "If the correct action is performed, will the user see that progress is being made?"

P. J. Smith, Stone, and Spencer (2006) suggested some additional, more detailed questions that can be asked to help the analyst consider how different cognitive processes influence performance (Card et al., 1983; Wickens et al., 2004):

- Selective Attention: What are the determinants of attention? What is most salient in the display? Where will the user's focus of attention be drawn? (See Bennett & Flach, 1992; Eckstein, 2004; Johnston & Dark, 1986; Lowe, 2003; Pashler, Johnston, & Ruthruff, 2001; Yantis, 1998.)
- Automaticity and Controlled Attentional Processes: Does the design support skill-based performance for routine scenarios and controlled, knowledge-based processes for novel situations? (See Logan, 2005; Rasmussen, 1983; Rasmussen et al., 1994.)
- Perception: How will perceptual processes influence the user's interpretation? How, for instance, will the proximity of various items on the screen influence judgments of "relatedness" as predicted by the Gestalt Law of Proximity? (See Bennett & Flach, 1992; Kohler, 1992; Tufte, 1983, 1990, 1997; Vicente, 2002; Watzman, 2003.)
- Learning and Memory: How will the user's prior knowledge influence selective attention and interpretation? Does the knowledge necessary to perform tasks reside in the world or in the user's mind? (See Baddeley, 1998, 1999; Grondin, 2005; Hester & Garavan, 2005; Hutchins, 1995; Norman, 2002; Ormrod, 2003; Radvansky, 2005; Scaife & Rogers, 1996; Seamon, 2005.)
- Information Processing, Mental Models, and Situation Awareness: What inferences/assumptions will the user make? What internal representations will support such processing? (See Endsley, 2003; Endsley & Garland, 2000; Gentner & Stevens, 1983; Johnson et al., 1981; Plous, 1993; Schaeken, 2005.)
- Design-Induced Error: How could the product design and the context of use influence performance and induce errors? (See Bainbridge, 1983; Beach & Connolly, 2005; Johnson et al., 1981; Larson & C. Hayes, 2005; Parasuraman & Riley, 1997; Reason, 1991, 1997; Roth, Bennett, & Woods, 1987; Sheridan, 2002; Skirka, Mosier, & Burdick, 1999; P. J. Smith & Geddes, 2003; P. J. Smith, McCoy, & Layton, 1997.)
- Motor Performance: Can the controls be used efficiently and without error? (See Jagacinski & Flach, 2003.)
- Group Dynamics: How will the system influence patterns of interaction among people? (See Baecker, Grudin, Buxton, & Greenber, 1995; Brehm, Kassin, & Fein, 1999; Forsyth, 1998; Levi, 2001; G. M. Olson & J. S. Olson, 2003.)

Thus, steps 1–3 as outlined above involve describing how a product (such as a DSS) can be used to achieve a specific goal. Step 4 involves predicting how people might perform when using this product, either successfully or unsuccessfully, as they try to achieve this goal. Thus, the "cognitive" part of this analysis is embedded only in step 4. Note also that the goal-subgoal hierarchy does not attempt to represent the further sequences of actions that could arise once the user starts down a failure path. This helps to keep the size of the hierarchy smaller, keeping the analysis task more manageable. (For some applications, it may be desirable to actually expand the goal-subgoal hierarchy to represent the complete path of a failure story.)

Finally, going beyond just the use of hierarchical decomposition techniques, Gordon and Gill (1997) noted that:

> CTA [Cognitive Task Analysis] methodologies differ along several dimensions. These include the type of knowledge *representation* or formalism used to encode the information obtained from the analysis, the *methods* used for eliciting or identifying expert (or novice) knowledge, and the *type of task and materials* used in the analysis. Typical representation formats include lists and outlines, matrices and cross-tabulation tables, networks, flowcharts, and problem spaces with alternative correct solution paths (pp. 133–134)

Thus, although the use of hierarchical goal-subgoal representations is one of the more popular approaches, a variety of other complementary approaches are labeled "cognitive task analyses" (Crandall, Klein, & Hoffman, 2006). These alternatives include the use of concept maps and conceptual graph analyses (Gordon & Gill, 1992) and different approaches to extracting and representing the knowledge of experts about critical incidents (Crandall et al., 2006; Klein, Caulderwood, & MacGregor, 1989). The ultimate goal of all of these approaches, however, is to better represent and predict how users' knowledge and control processes or cognitive processes will be applied in a given task environment. This task environment could be the existing world (a descriptive analysis) that is being studied in order to design improved support tools, or an envisioned world that incorporates some specific design proposal intended to improve performance (a predictive analysis).

Work-Domain Analyses

Cognitive task analyses tend to focus on how to support performance for an already identified task or goal. However, in many cases DSSs are designed to support performance in very complex environments (Bar-Yam, 2003, 2005) where it is unlikely that all of the possible scenarios that could arise will be predicted during the design process. Thus, there is also a need for design methodologies that support the development of robust designs that will be effective even when novel situations arise (Vicente, 1999).

To deal with this need, a complementary set of design methodologies have been developed under the label of "work-domain analysis" (Burns, Bizantz, & Roth, 2004; Burns & Hajdukiewicz, 2004; Burns & Vicente, 2001; Reising & Sanderson, 2002; Vicente, 2002; Vicente & Rasmussen, 1990, 1992). In general terms, the emphasis is on understanding and modeling the work domain or application area rather than simply modeling the performance of people on a prespecified set of tasks within this domain. The argument is that the domain analysis can identify the critical parameters or constraints relevant to successful performance in that domain and thus serve as the basis for a design that can handle both routine and novel (unanticipated) situations.

To make this point more concrete, as a very simple but illustrative example, Vicente (2000) contrasted verbal directions on how to travel from one point to another with the use of a map. In this analogy, the directions provide a very efficient but brittle description of how to accomplish the desired goal. The map requires more work (interpretation), but is more flexible in the face of obstacles that might render the directions unusable. Building on this analogy, Vicente noted:

> This decrease in efficiency [of maps vs. directions] is compensated for by an increase in both flexibility and generality. Like maps, work domain representations are more flexible because they provide workers with the information they need to generate an appropriate response online in real time to events that have not been anticipated by designers. This is particularly useful when an unforeseen event occurs because task representations, by definition, cannot cope with the unanticipated (p. 115)

The literature on work-domain analyses discusses the use of abstraction (means-end) hierarchies and aggregation (part-whole) hierarchies to complete a domain analysis, and then using the results of this analysis to guide the design of the DSS for supporting operator performance (Burns & Hajdukiewicz, 2004; Vicente, 1999; Vicente & Rasmussen, 1990). The basic logic of this approach is that, in order to design appropriate representations to assist an operator in completing tasks using some system, the complete and correct semantics of the domain must first be appropriately defined. Since such a design is concerned with supporting goal-oriented behaviors, one way to represent the semantics is with an abstraction (means-end) hierarchy that captures the "goal-relevant properties of the work domain" (Vicente & Rasmussen, 1990, p. 210). In addition, because the domains of interest are often quite complex, aggregation hierarchies may be useful to decompose the overall domain into a set of interrelated subparts.

Approaches to Design—Summary

Much of the work on methods for completing cognitive task analyses and work domain analyses has been stimulated by the challenges of developing DSSs to support performance on complex tasks. The description above provides a sense of the range of techniques that have been developed, and emphasizes the need to understand both the constraints of the application context and the nature of the agent(s) (person or people) involved. The insights provided by such analyses can be used to guide the design of the underlying functionality embedded within a DSS, and to guide the design of the interaction between a DSS and its user.

HUMAN PERFORMANCE ON DECISION-MAKING TASKS

There are several reasons why an understanding of human performance is important to the designer of a DSS. The first is that the motivation for developing such systems is to increase efficiency (reduce production costs or time) and/or to improve the quality of performance. Thus, it is important for the design team to be able to efficiently and effectively complete the initial problem definition and knowledge-engineering stages of a project, identifying areas in which improvements are needed. An understanding of human performance makes it possible to, in part, take a top-down approach to this, looking to see whether certain classic human performance phenomena (such as hypothesis fixation) are influencing outcomes in a particular application. A second motivation for understanding human performance is that a human-centered approach to the design of a DSS requires consideration and support of the user's skills (Garb, 2005). To do this effectively, knowledge of human performance (perception, learning and memory, problem solving, decision making, etc.) is essential. A number of these aspects of human performance are covered in previous chapters, so this chapter will just highlight some of the factors most relevant to the design of a DSS, and discuss how such factors are relevant to this design task.

Errors and Cognitive Biases

In very broad terms, human errors (Strauch, 2004; Wiegmann & Shappell, 2003) can be classified as slips and as mistakes (Norman, 1981). Slips arise through a variety of cognitive processes, but are defined as behaviors in which the person's actions do not match his intentions. Generally, this refers to cases where the person has the correct knowledge to achieve some goal but, due to some underlying perceptual, cognitive, or motor process, fails to correctly apply this knowledge. As Norman described it: "Form an appropriate goal but mess up in the performance, and you've made a slip" (1981, p. 106). Mistakes, on the other hand, refer to errors resulting from the accurate application of a person's knowledge to achieve some goal, but where that knowledge is incomplete or wrong.

DSSs are potentially useful for dealing with either of these sources of errors. If slips or mistakes can be predicted by the design team, then tools can be developed to either help prevent them, recover from them, or reduce their impacts.

Slips

Norman (2002) discussed six categories of slips. Knowing something about these different causes can help the designer to look for possible manifestations in a particular application area. The six categories, as defined in Norman, included:

- Capture errors, "in which a frequently done activity suddenly takes charge instead of (captures) the one intended" (p. 107);
- Description errors, where "the intended action has much in common with others that are possible" and "the internal description of the intention was not sufficient . . . often resulting in "performing the correct action on the wrong object" (pp. 107–108);
- Data-driven errors, where an automatic response is triggered by some external stimulus that triggers the behavior at an inappropriate time;
- Associative activation errors where, similar to a data-driven error, something triggers a behavior at an inappropriate time, but in this case the trigger is some internal thought or process;
- Loss-of-activation errors, or "forgetting to do something" (p. 109);
- Mode errors, or performing an action that would have been appropriate for one mode of operation for a system, but is inappropriate for the actual mode or state that the system is in.

Mistakes

As defined above, mistakes are due to incorrect knowledge (the rule, fact, or procedure that the person believes to be true is incorrect, resulting in an error) or incomplete (missing) knowledge.

Cognitive Biases

The literature on human error also provides other useful ways to classify errors in terms of surface level behavior or the underlying cognitive process. Many of these are discussed under the label of "cognitive biases" (Brachman & Levesque, 2004; Bradfield & Wells, 2005; Fraser, P. J. Smith, & J. W. Smith, 1992; Gilovich, Griffin, & Kahneman, 2002; Haselton, Nettle, & Andrews, 2005; Kahneman & Tversky, 2000; Plous, 1993; Poulton, 1989), including:

- Gambler's fallacy (Croson & Sundali, 2005). (Concluding there is a pattern in a series of events that is in reality a random sequence.) Gilovich, Vallone, and Tversky (1985) for instance, found that individuals believe they see streaks in basketball shooting even when the data show that the sequences are essentially random.
- Insensitivity to sample size (Bjork, 1999). (Failing to understand that the law of large numbers implies that the probability of observing an extreme result in an average decreases as the size of the sample increases.) Tversky and Kahneman (1974), for example, found that subjects believed that "the probability of obtaining an average height greater than 6 feet was assigned the same value for samples of 1000, 100 and 10 men." (p. 1127)
- Incorrect revision of probabilities. (Failure to revise probabilities sufficiently when data are processed simultaneously, or, conversely, revising probabilities too much when processing the data sequentially.)
- Ignoring base rates. (Failure to adequately consider prior probabilities when revising beliefs based on new data.)
- Use of the availability heuristic (Blount & Larrick, 2000). (Tversky (1982) suggested that the probability of some type of event is in part judged based on the ability of the person to recall events of that type from memory, thus suggesting that factors like recency may incorrectly influence judgments of probability.)
- Attribution errors. Jones and Nisbett (1971) described this by noting that "there is a pervasive tendency for actors to attribute their actions to situational requirements, whereas observers tend to attribute the same actions to stable personal dispositions." (p. 83)
- Memory distortions due to the reconstructive nature of memory (Blank, 1998; Handberg, 1995). (Loftus (1975) described processes that distort memories based on the activation of a schema as part of the perception of an event, and the use of that schema to reconstruct the memory of the event based on what the schema indicates should have happened rather than what was actually perceived originally. P. J. Smith, Giffin, Rockwell, and Thomas (1986) and Pennington and Hastie (1992) provided descriptions of similar phenomena in decision-making tasks.)

In recent years, there has been considerable controversy regarding the nature and validity of many of these explanations as "cognitive biases" (Fraser et al., 1992; Haselton, 2005; Koehler, 1996), suggesting that it is important to carefully understand the specific context in order to generate predictions as

to whether a particular behavior will be exhibited. Using the incorrect revision of probabilities as an illustration, Navon (1978) suggested that in many real-world settings the data aren't independent, and that what appears to be conservatism in revising probabilities in a laboratory setting may be the result of the subjects applying a heuristic or cognitive process that is effective in real-world situations where the data are correlated.

A more detailed example is provided by the popular use of the label "hypothesis fixation." This behavior refers to some process that leads a person to form an incorrect hypothesis and to stick with that hypothesis, failing to collect critical data to assess its validity or to revise it in the face of conflicting data. A variety of cognitive processes have been hypothesized to cause this behavior. One example is called "biased assimilation," in which a person judges a new piece of data as supportive of his hypothesis (increasing the level of confidence in the hypothesis) based simply on the consideration of whether that outcome could have been produced under his hypothesis. This contrasts with inferential processes based on a normative model that suggest that beliefs should be revised based on the relative likelihood of an outcome under the possible competing hypotheses, and that would, in the same circumstance, lead to a reduction in the level of confidence in his hypothesis.

Another example of a cognitive process discussed in the literature as relevant to hypothesis fixation is the so-called "confirmation bias." This phenomenon (described in Wason (1960) and in Mynatt, Doherty, and Tweney (1978)) is concerned with the person's data-collection strategy. For example, in a study asking subjects to discover the rules of particle motion in a simulated world, Mynatt et al. described performance by concluding that there was "almost no indication whatsoever that they intentionally sought disconfirmation." Later studies, however, have suggested that this "bias" is really an adaptive "positive test" strategy that is effective in many real-world settings, and that the unusual nature of the task selected by Mynatt et al. makes it look like an undesirable bias (Klayman & Ha, 1987). P. J. Smith et al. (1986) further suggested that, in real-world task settings where such a strategy might lead to confirmation of the wrong conclusion, experts often have domain-specific knowledge or rules that help them to avoid this strategy altogether, thus ensuring that they collect critical potentially disconfirming evidence.

Designer Error

Many of the error-producing processes discussed above appear to focus on system operators. It is important to recognize, however, that the introduction of a DSS into a system is a form of cooperative work between the users and the design-and-implementation team and that, like the users, the design team is susceptible to errors (Petroski, 1994). These errors may be due to slips or mistakes. In the case of mistakes, it may be due to inadequate knowledge of the application area (the designers may fail to anticipate all of the important scenarios) or incorrect knowledge. It may also be due to a failure to adequately understand or predict how the users will actually apply the DSS, or how its introduction will influence their cognitive processes and performances. In addition, sources of errors associated with group dynamics may be introduced. Whether such errors are due to a lack of sufficient coordination, where one team member assumes that another is handling a particular issue, or due to the influence of group processes (Brehm et al., 1999; Forsyth, 1998; Janis, 1982; Levi, 2001; McCauley, 1989; Tetlock, 1998; Tetlock, Peterson, McQuire, Chang, & Feld, 1992) on design decisions, the result can be some inadequacy in the design of the DSS, or in the way the user and the DSS work together. (Note that such group processes are also important potential sources of errors when system operators work together as part of teams, with or without technological support.)

Systems Approaches to Error

Reason (1991, 1997) and Wiegmann, Zhang, Von Thaden, Sharma, and Gibbons (2004) cautioned designers against fixating on the immediately preceding "causes" of an accident when trying to prevent future occurrences. They remind us that, in many system failures, a number of conditions must coincide in order for the failure to occur. This presents a variety of leverage points for preventing future occurrences, many of which are preemptive and thus remote from the actual accident or system failure. Many of these changes focus on preventing the conditions that could precipitate the system failure, rather than improving performance once the hazardous situation has been encountered.

Errors and Cognitive Biases—Implications for Design

As described at the beginning of this section, the value of this literature to the designer is the guidance it provides in conducting the initial knowledge-engineering studies that need to be completed in order to identify opportunities for improvement in some existing decision-making process that is to be supported by a new DSS. Familiarity with this literature is also critical to the designer to ensure that the introduction of the DSS doesn't result in new design-induced errors. This literature also serves to emphasize two additional considerations:

- The occurrence of errors are often due to the co-occurrence of a particular problem-solving strategy with a given task environment that is "unfriendly" to that strategy (e.g., situations in which the strategy is not sufficiently robust). Strategies that were adaptive in one setting may no longer be adaptive in the newly designed environment. The potential for such a negative transfer of learning needs to be considered as part of the design.
- For routine situations, people tend to develop expertise that lets them employ knowledge-rich problem-solving strategies (Crandall et al., 2006; Klein, 1993; Kolodner, 1993; Newell, 1990) that avoid the errors that could be introduced by certain general problem-solving strategies. However, in many system designs, people are expected to act as the critical safety net during rare, idiosyncratic events that the design team has failed to anticipate. These are exactly the situations in which people have to fall back on their general problem-

solving strategies, and are thus susceptible to the potential errors associated with these weak problem-solving methods.

In addition, this review emphasizes the need to take a broad systems perspective, recognizing that the design and implementation team are a potential source of error, and understanding that, if the goal is to prevent or reduce the impact of errors, then the solution is not always to change performance at the immediate point where an error was made. It may be more effective to change other aspects of the system so that the potentially hazardous situation never even arises.

Finally, the emphasis on designer error implies that the design process for a DSS needs to be viewed as iterative and evolutionary. Just because a tool has been fielded, it is not safe to assume that no further changes will be needed. As Horton and Lewis (1991) discussed, this has organizational implications (e.g., ensuring adequate communications regarding the actual use of the DSS, budgeting resources to make future revisions), as well as architectural implications (developing a system architecture that enables revisions in a cost-effective manner).

Human Expertise

As discussed in previous chapters, the nature of human information processing imposes a variety of constraints that influence how effectively a person processes certain kinds of information. These include memory, perceptual, and information-processing constraints. As a result, there are certain decision tasks in which the computational complexity or knowledge requirements limit the effectiveness of unaided individual human performance.

From a design perspective, these information-processing constraints offer an opportunity. If important aspects of the application are amenable to computational modeling, then a DSS may provide a significant enhancement of performance, either in terms of efficiency or the quality of the solution. (Note that, even if development of an adequate computational model is not feasible, there may be other technological improvements, such as more effective communications environments, that could enhance performance. However, this chapter is focusing on DSSs that incorporate active information-processing functions by the software.)

A good example of this is flight planning for commercial aircraft. Models of aircraft performance considering payload, winds, distance, and aircraft performance characteristics (for the specific aircraft as well as the general type of aircraft) are sufficiently accurate to merit the application of DSSs that use optimization techniques to generate alternative routes and altitude profiles to meet different goals in terms of time and fuel consumption (P. J. Smith, McCoy, & Layton, 1997). This example also provides a reminder, however, that it is not just human limitations that are relevant to design. It is equally important to consider human strengths, and to design systems that complement and are compatible with users' capabilities. In flight planning, for instance, this includes designing a system that allows the person to incorporate his judgments into the generation and evaluation of alternative flight plans, considering the implications of uncertainty in the weather, air traffic congestion, and other factors that may not be incorporated into the model underlying the DSS.

This cooperative systems perspective has several implications. First, the designer needs to have some understanding of when and how the person should be involved in the alternative generation and selection processes. This requires insights into how the person makes decisions, in terms of problem-solving strategies as well as in terms of access to the relevant knowledge and data, and how the introduction of a DSS can influence these problem-solving processes. It also implies that the strengths underlying human perceptual processes need to be considered through display and representation-aiding strategies (Burns & Hajdukiewicz, 2004; Jones & Schkade, 1995; Kleinmuntz & Schkade, 1993; Larkin, 1989; Tufte, 1997) in order to enhance the person's contributions to the decision-making process by making important relationships more perspicuous or salient.

Descriptive Models

The literature on human problem solving and decision making provides some very useful general considerations for modeling human performance within a specific application. This literature, which is reviewed in more detail in previous chapters, covers a variety of descriptive models of problem solving. This includes the use of heuristic search methods (Clancey, 1985; Dasgupta, Chakrabarti, & Desarkar, 2004; Michalewicz & Fogel, 2004; Michie, 1986; Rayward-Smith, Osman, Reeves, & G. Smith, 1996; Russell & Norvig, 1995). This modeling approach (Newell & Simon, 1972) conceptualizes problem solving as a search through a space of problem states and problem-solving operations to modify and evaluate the new states produced by applying these operations. The crucial insight from modeling problem solving as search is an emphasis on the enormous size of the search space resulting from the application of all possible operations in all possible orders. In complex problems, the size of the space precludes exhaustive searching of the possibilities to select an optimal solution. Thus, some heuristic approach that satisfices, such as elimination by aspects (Tversky, 1972), is needed to guide a selective search of this space, resulting in the identification of an acceptable (if not optimal) solution.

Another aspect of problem solving that Newell and Simon (1972) noted when they formulated the search paradigm was the importance of problem representation. Problem representation issues emphasize that a task environment is not inherently objectively meaningful, but requires interpretation for problem solving to proceed. Some interpretations or representations are more likely to lead to successful solutions than others (for certain agents). In fact, an important component of expertise is the set of features or the representation used to characterize a domain (Lesgold et al., 1988), sometimes referred to as "domain ontology."

Task-specific problem spaces allow problem solvers to incorporate task- or environment-induced constraints, focusing attention on just the operations of relevance to preselected goals (Sewell & Geddes, 1990; Waltz, 1975). For such task-specific problem solving, domain-specific knowledge (represented in computational models as production rules, frames, or some other knowledge representation) may also be incorporated to increase search efficiency or effectiveness (Dechter, 2003;

Ghallib, Nau, & Traverso, 2004; Laird et al., 1987; Shalin, Geddes, Bertram, Szczepkowski, & DuBois, 1997).

These models, based on symbolic reasoning, have been developed for specific generic tasks like diagnosis or planning (Chandrasekaran, 1988; Ghallib et al., 2004; Gordon, 2004; G. A. Miller et al., 1960; Mumford, Schultz, & Van Doorn, 2001; Nareyek, Fourer, & Freuder, 2005). For example, computational models of planning focus on the use of abstraction hierarchies to improve search efficiency (Sacerdoti, 1974), dealing with competing and complementary goals (Wilensky, 1983), and mixed top-down/bottom-up processing (B. Hayes-Roth & F. Hayes-Roth, 1979) to opportunistically take advantage of data as it becomes available. More recent developments have sought to deal with planning in stochastic environments (Madani, Condon, & Hanks, 2003; Majercik & Littman, 2003).

In contrast to such sequential search models, models based on case-based reasoning, recognition-primed decision making, or analogical reasoning (Crandall et al., 2006; Klein, 1993; Kolodner, 1993; Riesbeck & Schank, 1989) focus on prestructured and indexed solutions based on previously experienced situations. These modeling approaches suggest that human experts in complex operational settings rarely describe their cognitions as sequential searches to construct alternative solution states, but rather as recognition processes that match features of the situation to prototypical, preformulated response plans (Zuboff, 1988; Rasmussen, 1983). In complex dynamic domains (Oliver & Roos, 2005) like aviation or firefighting, these preformulated plans incorporate a great deal of implicit knowledge reflecting the constraints of the equipment, the environment, other participants in the work system, and previous experience. Over time, solutions sensitive to these constraints result in familiar, accepted methods which are then triggered by situational cues and modified in small ways as needed to deal with the specific scenario. However, it is important to recognize that such recognition-based processes can cause familiar task features to invoke only a single interpretation, resulting in hypothesis fixation or hindering creative departure from normal solutions. This limitation can be critical when the task environment contains a new, unexpected feature that requires a new approach.

Normative Optimal Models

In addition to these descriptive models, a large amount of literature characterizes human decision making relative to various normative models based on optimal processes, which help to emphasize important factors within the task and task environment that should be considered in making a decision in order to arrive at a better decision. These normative models are based on engineering models such as statistical decision and utility theory, information theory, and control theory (Jagacinski & Flach, 2003; Rouse, 1980; Sheridan & Ferrell, 1974) By contrasting human performance with optimal performance on certain tasks and emphasizing the factors that should influence decision making in order to achieve a high level of performance, this literature helps the designer to look for areas where human performance may benefit from some type of DSS.

Finally, the literature on human expertise (Charness, Feltovich, Hoffman, & Ericsson, 2006) emphasized the ability of people to learn and adapt to novel situations, and to develop skeletal plans that guide initial decisions or plans, but that are open to adaptation as a situation unfolds (Geddes, 1989; Suchman, 1987; Wilensky, 1983). The literature also emphasizes variability, both in terms of individual differences and in terms of the ability of a single individual to use a variety of decision-making strategies in some hybrid fashion in order to be responsive to the idiosyncratic features of a particular scenario.

Human Expertise—Implications for Design

An understanding of the literature on human expertise is of value to the designer in a number of different ways. First, in terms of the initial problem-definition and knowledge-engineering stages, familiarity with models of human problem solving and decision making can help guide the designer in looking for important features that influence performance within that application. Second, in an effort to provide cognitive compatibility with the users, the design of many of the technologies underlying DSSs is guided by these same computational models of human performance. Third, even if the underlying technology isn't in some sense similar to the methods used by human experts in the application, the designer needs to consider how the functioning of the DSS system should be integrated within the user's decision-making processes.

In terms of some specific emphases, the above discussion leads to some suggestions:

- Don't fixate on active DSS technologies as the only way to improve system performance. Enhancing human performance through changes in procedures, improvements in passive communication tools, better external memory aids, and so on may be more cost-effective in some cases. In addition, such changes may be needed to complement an active DSS in order to make its use more effective.
- Look for ways in which the direct perception of ecological constraints allows people to perform expertly (Flach, Hancock, Caird, & Vicente, 1995). The ability to perceive these critical parameters or constraints directly may make what seems like a very difficult information-processing task much less demanding.
- Related to the consideration of ecological constraints was the earlier suggestion that problem representation has a strong influence on how easily a problem solver can find a good solution. This issue of perspicuity (salience of the problem solution), however, is dependent not only on the characteristics of the task and task environment, but also on the nature of the problem solver. Thus, in order to enhance human performance as part of the decision-making process, it is important to consider alternative ways of representing the situation that will enable human perceptual and cognitive processes to work more effectively. In addition, problem representation can affect the performance of the designer. Consequently, alternative problem representations need to be generated as part of the design process to help the designer think more effectively.
- Consider the applicability of normative optimal models of performance for the task in order to focus attention on factors that should be influencing current performance, and to

contrast optimal strategies with the strategies that people are actually using in response to these task-determined factors. Aspects of these normative models may also be appropriate for more direct inclusion in the DSS itself, even though the strategies used by people do not fully reflect the optimal processes highlighted by these normative models. In addition to considering the task structure highlighted by normative optimal models, use knowledge of the variety of different descriptive models of human problem solving and decision making to guide knowledge-engineering efforts, making it easier and more efficient to understand how people are currently performing the tasks. Thus, all of these models of decision making represent conceptual tools that help the designer to more effectively and efficiently understand performance in the existing system, and to develop new tools and procedures to improve performance.

- Whether the DSS is designed to process information "like" the user at some level, or whether it uses some very different processing strategy, the user needs to have an appropriate and effective mental model of what the system is and is not doing (Kotovsky, J. R. Hayes, & Simon, et al., 1985; Lehner & Zirk, 1987; Nickerson, 1988; Zhang, 1997; Zhang & Norman, 1994). To help ensure such cognitive compatibility, the designer needs to understand how people are performing the task.
- Consider design as a prediction task, trying to predict how users will interact with the DSS system, and how it will influence their cognitive processes and performances.

Additional Cognitive Engineering Considerations

The preceding sections focused on our understanding of human performance on decision-making tasks and the implications of that literature for design. Below, some additional considerations based on studies of the use of DSSs are presented.

The Human Operator as Monitor

Numerous studies make it clear that, given the designs of DSSs for complex tasks must be assumed to be brittle, there is a problem with designs that assume that human expertise can be incorporated as a safety net by asking a person to simply monitor the output of the DSS, with the responsibility for overriding the software if a problem is detected. One problem with this role is that, in terms of maintaining a high level of attentiveness, people do not perform well on such sustained attention tasks (Meister & Enderwick, 2002). As Bainbridge (1983) noted:

> We know from many 'vigilance' studies (Mackworth, 1950/1961) that it is impossible for even a highly motivated human being to maintain effective visual attention towards a source of information on which very little happens, for more than about half an hour. This means that it is humanly impossible to carry out the basic function of monitoring for unlikely abnormalities. (p. 777)

A related issue is the problem of loss of skill. As Bainbridge (1983) further noted, if the operator has been assigned a passive monitoring role, he "will not be able to take over if he has not been reviewing his relevant knowledge, or practicing a crucial manual skill" (p. 778). Thus, a major challenge for retaining human expertise within the system is "how to maintain the effectiveness of the human operator by supporting his skills and motivation" (p. 778).

Complacency and Overreliance

Studies reported by Parasuraman and Riley (1997) and Parasuraman (2000) introduced further concerns about assigning the person the role of critiquing the computer's recommendations before acting. These studies discuss how the introduction of a DSS system can lead to overreliance by the human user when the software is generating the initial recommendations (Metzger & Parasuraman, 2005; Skirka et al., 1999).

Excessive Mental Workload

Designs that relegate the person to the role of passive monitor run the risk of a vigilance decrement due to insufficient engagement and mental workload. At the other extreme, designers must recognize that "clumsy automation" that leaves the person with responsibility for difficult parts of the task (such as coping with an emergency), but that adds additional workload (Kushleyeva, F. Salvucci, & Schunn, 2005) due to the awkward interactions now required to access information and functions embedded in the DSS (such as navigating through complex menus to view certain information), can actually impair performance because of the added mental workload of interacting with the DSS (Wiener & Nagel, 1988). One line of research that is now getting increased attention is the potential to design adaptive systems that can monitor the human operator using neurophysiological measures that attempt to detect problems with alertness, mental workload, or attention (Caggiano & Parasuraman, 2004; Freeman, Mikulka, & Scerbo, 2004; Grier, Warm, & Dember, 2003; Luo, Greenwood, & Parasuraman, 2001; Mikulka, Scerbo, & Freeman, 2002). This work seeks to "determine whether a biocybernetic, adaptive system could enhance vigilance performance," with the goal of improving "monitoring performance on critical activities such as air traffic control and radar and sonar operation" (Mikulka et al., 2002, p. 654), or offloading work to software or other people when the computer detects that excessive attentional demands are being placed on an individual.

Lack of Awareness or Understanding

Even if the person is sufficiently engaged with the DSS and the underlying task, designers need to consider how to ensure that the user has an accurate mental model of the situation, and of the functioning and state of the DSS (Billings, 1997; Larkin & Simon, 1987; Mitchell & R. A. Miller, 1986; Roth et al., 1987; Sarter & Woods, 1993). If the person does not have such an understanding, then it may be difficult for him to intervene at appropriate times or to integrate the computer's inputs into his own thinking appropriately (Geddes, 1997b). This concern has implications for selection of the underlying technology and conceptual model

for a system, as well as for the design of the visual or verbal displays intended to represent the state of the world and the state of the software for the user, including explanations of how the DSS has arrived at its recommendations (Clancey, 1983; Hasling, Clancey, & Rennels, 1984; Nakatsu & Benbasat, 2003).

Lack of Trust and User Acceptance

As outlined above, overreliance can be a problem with certain assignments of roles to the person and the computer. At the other extreme, lack of trust or acceptance of the technology can eliminate or reduce its value (Clarke, Hardstone, Rouncefield, & Sommerville, 2006; Hollnagel, 1990; Muir, 1987). This lack of acceptance can result in outright rejection of the DSS (in which case it either is not purchased or not used), or in a tendency to underutilize it for certain functions. It is important to note that this lack of acceptance can be due to resistance to change (Cartwright & Zander, 1960; Forsyth, 1998; Levi, 2001) even if there is no intrinsic weakness in the DSS, due to general beliefs held by the operators (rightly or wrongly) about how the software will influence their lives, or due to beliefs about how well such a DSS can be expected to perform (Andes & Rouse, 1992; J. Lee & See, 2004; Marsh & Dibben, 2003; Muir & Moray, 1996; Riegelsberger, Sasse, & McCarthy, 2005).

Active Biasing of the User's Cognitive Processes

Complacency is one way in which a DSS can influence the person's cognitive processing. Studies have also shown, however, that the use of a DSS can also actively alter the user's cognitive processes (Beach & Connolly, 2005; Larson & C. Hayes, 2005; P. J. Smith, McCoy, & Layton, 1997). The displays and recommendations presented by the software have the potential to induce powerful cognitive biases, including biased situation assessment and failures by the user to activate and apply his expertise because normally available cues in the environment are no longer present. The net result is that the person fails to exhibit the expertise that he would normally contribute to the decision-making task if working independently, not because he lacks that expertise but because the computer has influenced his cognitive processes in such a way that this knowledge is never appropriately activated. Studies have shown that these biasing effects can induce practitioners to be 31% more likely to arrive at an incorrect diagnosis on a medical decision-making task (Guerlain et al., 1996) and 31% more likely to select a very poor plan on a flight-planning task (P. J. Smith, McCoy, & Layton, 1997).

Distributed Work and Alternative Roles

Much of the literature focuses on the interactions between a single user and the DSS. Increasingly, however, researchers and system developers are recognizing that one approach to effective performance enhancement is to think in terms of a distributed work paradigm (Schroeder & Axelsson, 2005; P. J. Smith, McCoy, & Orasanu, 2001), in which the software may be one "agent" in this distributed system (Geddes & Lizza, 1999), or the software may be viewed primarily as a mediator to support human-human interactions within virtual or distributed teams (Baecker et al., 1995; Bowers et al., 2006; Caldwell, 2005; Carroll, Neale, Isenhour, & McCrickard, 2003; Handy, 1995; Hertel, Geister, & Konradt, 2005; Hinds & Kiesler, 2005; Hinds & Mortensen, 2005; Hutchins, 1990, 1995; Katz, Lazer, Arrow, & Contractor, 2004; Lurey & Raisinghani, 2001; G. M. Olson & J. S. Olson, 1997, 2003; Orasanu & Salas, 1991; Rasmussen et al., 1991; Salas, Bowers, & Edens, 2001; Salas & Fiore, 2004; Smith, McCoy, & Layton, 1997). Such distributed systems can now be found in the military, in medicine, in aviation and a variety of other application areas.

Sycara and Lewis (2004) suggested that:

> There are three possible functions that software agents might have within human teams:
>
> 1. Support the individual team members in completion of their own tasks
> 2. Assume the role of a (more or less) equal team member by performing the reasoning and tasks of a human teammate
> 3. Support the team as a whole. (p. 204)

and that "teams could be aided along four general dimensions: accessing information, communicating, monitoring, and planning" (p. 227). In considering the design and use of DSSs for such applications, all of the more traditional factors must be considered. In addition, questions regarding the impact of the technology on team situation awareness (Endsley, 2003), rapport, trust, and communication become paramount. It is also important to think of a broader set of design parameters, as distributed approaches to work open up the potential to use a variety of different architectures for distributing the work in terms of the locus of control or responsibility and the distribution of knowledge, data, and information-processing capabilities (Griffith, Sawyer, & Neale, 2003; Sheridan, 1997; Sheridan, 2002; P. J. Smith, McCoy, Orasanu, et al., 1997; P. J. Smith, Billings, et al., 2000; P. J. Smith, Beatty, Spencer, & Billings, 2003). Furthermore, DSSs may be necessary to make effective certain architectures for distributing the work.

Organizational Failures

The discussions above focus on design-induced errors made by the system operators. It is equally important for the design team to recognize that part of the design process is ensuring that the management of the organization into which the DSS is introduced will provide an effective a safety net. This means that the design team needs to ensure that the organization has established effective procedures to detect significant problems associated with the introduction and use of the DSS, and that such problems are communicated to the levels of management where responsibility can and will be taken to respond effectively to these problems (Horton & Lewis, 1991).

CASE STUDIES

Below, case studies are presented focusing on the designs of three quite different DSSs. Within these case studies, a number

of the issues raised earlier are discussed in more detail within the context of real-world applications, along with presentations of design solutions that provide concrete illustrations of how to deal with those issues.

Case Study A—Distributed Work in the National Airspace System: Ground Delay Programs

Earlier in this chapter, we indicated that a distributed-work system can be characterized in terms of the assignment of control or decision-making authority in relationship to the distribution of knowledge and data. The Federal Aviation Administration's (FAA's) Enhanced Ground Delay Program provides a very informative example for illustrating the significance of these parameters when designing a distributed system.

The Application Area

The FAA's Enhanced Ground Delay Program (GDP) has been in operation since 1998. Its goal is to help match the arrival demand at an airport with the available capacity, where demand could exceed capacity because of a weather constraint at that airport, a closed runway, and so on.

The strategy underlying the use of GDPs is to delay departures to the constrained airport, holding these departures on the ground when that airport's arrivals are predicted to exceed capacity. Such a strategy has potential benefits for the FAA in terms of reducing air traffic controller workload (reducing the need for the use of airborne holding or some other air traffic control tactic). It also has potential benefits for the NAS Users, as it can be more economical to delay departures, holding these flights on the ground (reducing fuel consumption due to airborne holding as well as reducing the potential for diversions).

The Original Approach to Distributing Work in a GDP

In principle, there are a variety of different ways to implement GDPs. Prior to 1998, the FAA used GDPs, but did so using an approach that was much less effective than the currently used procedure. Under this original paradigm, traffic managers at the FAA's Air Traffic Control Systems Command Center (ATCSCC) would set the airport arrival rate (AAR) for an airport, and then a DST would automatically assign an arrival slot to each specific flight, which would then be held on the ground until it was time to depart to make its assigned arrival slot. NAS Users had one opportunity to swap flights in a batch mode (e.g., requesting all of the swaps at one time). This limited their ability to monitor the development of the situation over time and adapt as new information became available. In addition, although NAS Users were asked to cooperate in order to make this program work, there was actually a disincentive for them to provide the FAA with information about cancellations or potential delays. If the cancellation of a flight was reported to the FAA, the NAS User lost its assigned arrival slot; if it was reported that a flight was going to miss its departure time (because of a mechanical problem, for instance), that flight could be assigned an even greater delay. As a result, NAS Users often chose to withhold this information from the FAA, resulting in unused arrival slots (wasted capacity).

An Alternative Approach to the Design of GDPs

Given these limitations, a joint industry-FAA program evolved to identify and implement a different approach to GDPs. This Collaborative Decision Making program explored methods for still achieving the FAA's primary goal (matching arrival rates to airport capacity), while giving NAS Users the flexibility to better achieve their business goals.

One approach would have been to have NAS Users provide the FAA with all of the knowledge and data about their business concerns relevant to decisions about which flights to assign to which arrival slots. In principle, the FAA traffic managers could have then determined "good" slot allocations for each NAS User, and assigned flights to slots accordingly. Such a decision would have had to somehow take into consideration the competitive nature of this situation and through some decision process determine an "equitable" solution.

It was decided that such an approach would produce unrealistic demands in terms of cognitive complexity and mental workload for these traffic managers, as they would have had to master the relevant expertise of the dispatchers working for each NAS User and to somehow use that knowledge to generate equitable slot assignments. (Dispatchers work for airlines and other general aviation operations that are required to, or choose to, abide by Part 121 regulations of the U.S. Federal Air Regulations, sharing responsibility for each flight with the pilots. They complete the preflight planning—determining the flight's route, departure time, fuel load, and so on—and are responsible for following that flight while it is airborne.)

Strategy for distributing work. Instead, the CDM program developed a new architecture for distributing the work associated with GDPs. This architecture uses a classic systems engineering approach for dealing with complexity, decomposing the overall plan for a GDP into a set of nearly independent subplans that can be developed by different individuals and that, when completed by each of these individuals, achieves the desired goals (assuming the plan is successfully implemented). The significance of the qualifier "nearly" should not be overlooked in terms of its importance, however. The reality is that there are cases where interactions among these different individuals are required, so the system design has to include mechanisms that support such interactions in order to ensure that the appropriate perspectives are considered in setting up a GDP, and that exceptions can be made for individual flights when that is required to deal with some overriding consideration.

Like the original GDP, this Enhanced GDP starts with the assumption that a "neutral resource broker" is needed because, in situations where a GDP is called for, the NAS Users are competing for a limited resource (arrival slots at the constrained airport). The critical difference, however, is the application of the following principle: "The "referee" (the FAA) should only control the process at the level of detail necessary to achieve its goals concerned with safety and throughput, thus allowing NAS Users the flexibility to try to achieve their own goals to the ex-

tent possible subject to this higher-level constraint." Thus, the critical change is the level of abstraction at which ATCSCC (the neutral referee) controls the process. Instead of determining which specific flights are allocated to specific arrival slots, ATCSCC determines the start and stop times for the GDP, the arrival rate for the constrained airport, and the departure airports to be included in the GDP, and then assigns arrival slots to specific NAS Users (such as a specific airline) rather than to specific flights (using a software tool implemented by Metron Aviation called the Flight Schedule Monitor or FSM, shown in Fig. 29.1).

The NAS User can then decide for itself which of its flights to assign to one of its arrival slots. An airline might, for instance, have flights from Dallas, Chicago, and Miami all scheduled to fly to LaGuardia in New York, all scheduled to arrive between 1800–1815Z, and all included in a GDP for LaGuardia. If that airline was given "ownership" of an arrival slot at 1855Z in a GDP for LaGuardia, it could assign that slot to any one of these three flights based on their relative importance to that airline's business concerns—considering factors such as passenger connections, aircraft requirements for later flights out of LaGuardia, crew time restrictions, and aircraft maintenance schedules.

Ration-by-schedule (RBS). In order to accomplish this new process, there are several requirements. First, a traffic manager at ATCSCC has to set the start and end times and the airport arrival rate, and determine which departure airports to include. To do this, ATCSCC solicits input from the relevant FAA facilities and the NAS Users. This interaction takes the form of a teleconference that is held every two hours, during which ATCSCC can propose the use of a GDP and provide an opportunity for input from each of the participants. This decision-making process is supported by a simulation or "what-if" function in FSM that allows the user to input a set of parameters (proposed arrival rate, start and end times, and included departure airports) and view the predicted impact in terms of average and maximum delays, and the potential for a spike in demand at the airport after the end time for the GDP. Window 3 in Fig. 29.1 shows such a demand pattern for SFO in one-hour

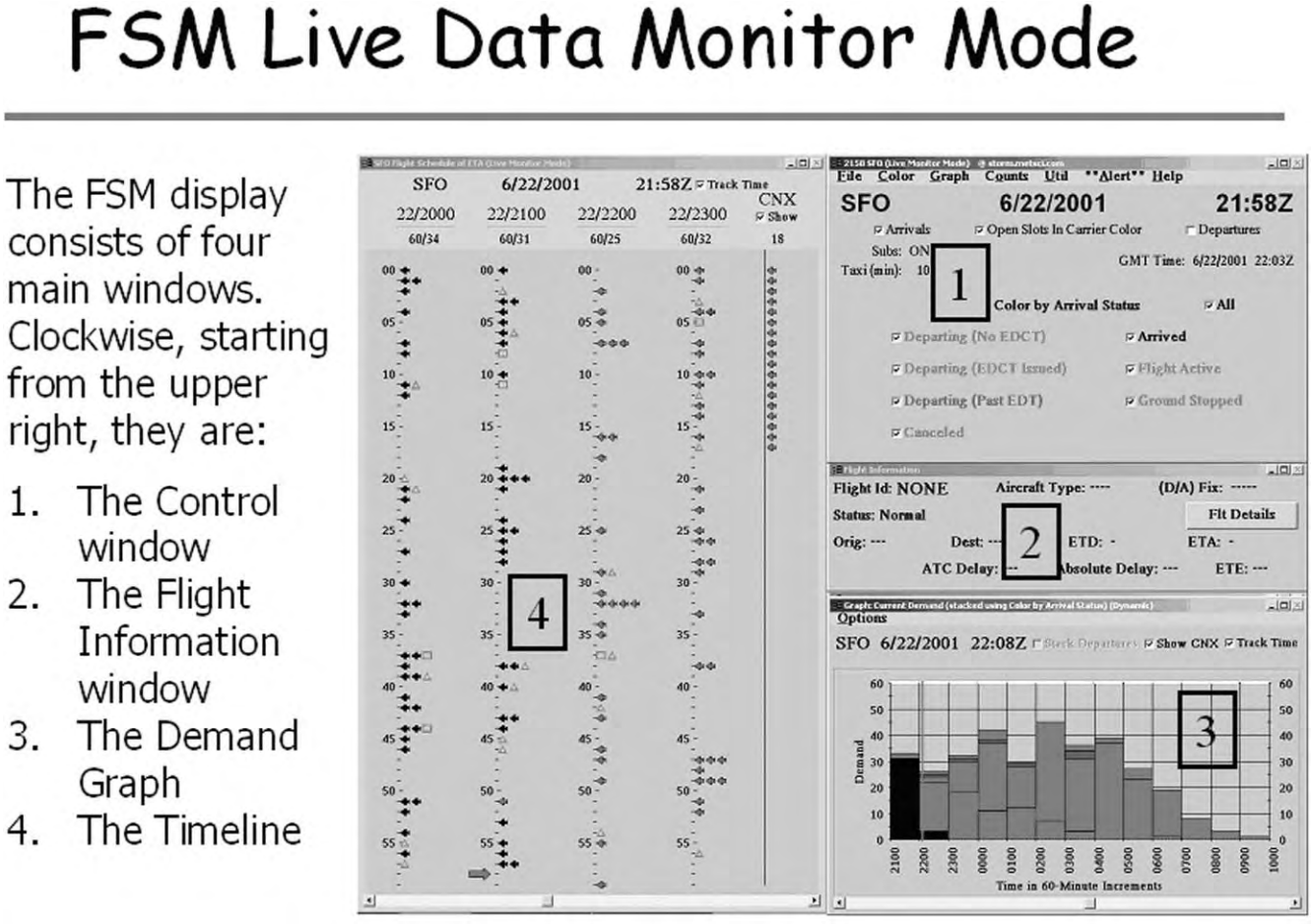


FIGURE 29.1. Screen display from FSM . (Window 1 controls the display of information in FSM; Window 2 provides access to information about specific flights included in the GDP; Window 3 provides an indication of the predicted arrival rate, with the display changing based on the parameters set for the GDP, thus supporting "what-iffing"; Window 4 indicates the predicted arrival times for individual flights.) (Source: Metron Aviation, Inc.)

time bins. The predicted peak demand, for instance, is 45 flights arriving during the 0200Z hour.

After considering these inputs, the ATCSCC traffic manager sets the parameters for the GDP, and lets FSM determine the arrival slot allocations for each NAS User. These allocations are based on a "Ration By Schedule" (RBS) logic. The number of arrival slots each user gets is proportional to its share of the flights that are included in the GDP that were originally scheduled to arrive at the constrained airport. The slots "owned" by that NAS User are initially assigned to its aircraft based on the order of their originally scheduled arrival times (see Fig. 29.2). Thus, if that NAS User originally has 20 flights out of a total of 80 flights that are included in the GDP and are scheduled to arrive at the constrained airport from 1300–1400Z (for a GDP from 1200Z-1900Z), and if the GDP requires this total of 80 to be reduced to 40, then that NAS User gets 10 slots from 1300–1400Z. As a default, the first 10 of its included flights scheduled to arrive after 1300Z are assigned these 10 slots. The arrival times for the remaining 10 flights are pushed back to slots after 1400Z.

Note that, to make this logic work, the GDP needs to cover a period in which, during the GDP or just after its end, there are valleys in the predicted demand that fall below the reduced airport arrival rate (AAR), so that the program doesn't create a huge peak in arrival demand at the end of the GDP. The parameters defining the GDP, along with the slot allocations, are then disseminated to the NAS Users.

Slot swapping. Abstractly, the strategy of controlling the process at a more abstract level (by setting an AAR) and the strategy of using the RBS algorithm to ration slots to NAS Users are critical, as they make it possible to distribute the overall work as relatively independent subtasks. The task of the FAA (the neutral referee) is to set the high-level control parameters. The NAS Users can then work within the resultant constraints to implement plans for their own fleets that are most effective from a business perspective.

The NAS Users accomplish this by swapping flights among assigned arrival slots. Each NAS User can do this by swapping flights among its own slots (see Fig. 29.3), or by making a swap with a slot "owned" by another NAS User, if such a swap is available.

Traditionally, decisions about what to do with a particular flight were handled by the individual dispatcher responsible for that flight. However, this new process involves making global decisions that consider tradeoffs among a number of flights. As a result, instead of requiring some type of potentially time-consuming collaboration among the affected dispatchers in that NAS User's Flight Operations Center, the responsibility for managing participation in a GDP has been assigned to a single senior dispatcher (often a new position referred to as the "Air Traffic Control" or "ATC Coordinator").

Most ATC Coordinators manage the GDP for their fleets using another set of DSSs that allow the ATC Coordinator to:

- Test alternative GDPs (GDPs with different parameters) to see which alternative is best from his fleet's perspective. (This information can be used for discussions during the planning teleconference in which the GDP is proposed, providing input to ATCSCC regarding the GDP parameters that should be used.)
- Swap arrival slots among different flights in the fleet either manually or though an automated process. (It is not unusual to make swaps involving a hundred flights based on passenger connections. This requires a DST that uses an algorithm to estimate the relative importance of the different flights. The ATC Coordinator can, however, constrain or override the solution recommended by the DST.)
- Test for the impact of strategies for canceling flights or delaying their departure times.

Figure 29.3 illustrates swapping flights belonging to the same NAS User. One very important variation of this feature is canceling a flight and then using its slot to swap with another flight. This is important because this provides the incentive for the NAS user to provide the FAA with critical information:

> According to the "rules of the game" for GDPs (P. J. Smith et al., 2003), when the NAS User cancels a flight, that User gets to keep its arrival slot and can swap it just like any other slot under its control. In order to make swaps with canceled flights, however, the User must inform the FAA of the cancellation. (accomplished through digital software communications)

This swapping process is further enhanced by a procedure that allows swapping among the different NAS Users. This procedure is referred to as "Slot-Credit Substitution."

Slot-Credit Substitution is best understood in the context of making a decision to cancel a flight. Suppose Airline ABC has flights that arrive every hour on the hour from LGA (LaGuardia

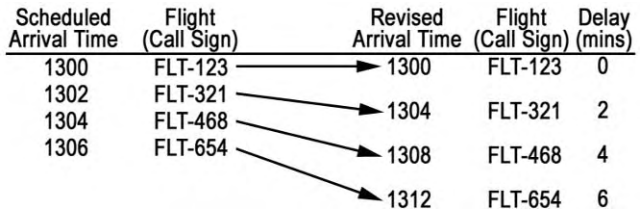


FIGURE 29.2. Arrival slot allocation using the Ration-By-Schedule algorithm. (Assume that the arrival airport can normally handle one arrival every two minutes. If the GDP reduces the arrival rate by 50% for all arrivals, then the flights are assigned revised arrival slots that keep the original ordering, but show increasing delay over time.)

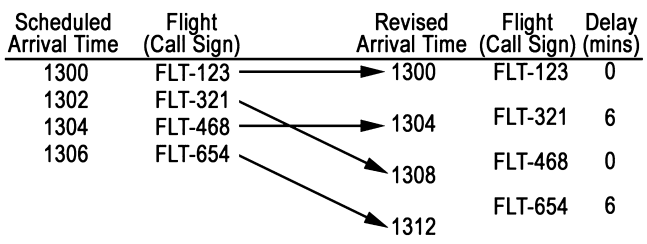


FIGURE 29.3. Arrival slot allocation after swapping FLT-321 and FLT-468. (Note that FLT-468 now has 0 minutes of delay instead of its original 4 minutes of delay.)

Airport) to ORD (Chicago O'Hare). As shown in Fig. 29.4, these flights are scheduled to arrive at ORD at 1000, 1100, 1200, and 1300Z.

Suppose a GDP is issued for ORD that delays these flights, assigning revised arrival times (slots) of 1030, 1200, 1300 and 1400Z, respectively. In principle, each of these flights is still capable of departing on time and therefore arriving at its originally scheduled arrival time if a suitable swap could be found to eliminate its delay due to the GDP. Thus the Earliest Runway Time of Arrival (ERTA) for each of these flights is the same as its originally scheduled arrival time. Each flight could, in principle, therefore be swapped to move its arrival to its ERTA. However, it could not be swapped earlier than its ERTA.

Given this situation, the ATC Coordinator for Airline ABC would be willing to cancel Flight 1 (and put the passengers on the next flight) if this would significantly reduce the delay on later flights. In the case shown in Fig. 29.4, however, even if he cancels Flight 1 the slot released is too early for Flight 2 (a flight with an 1100Z ERTA is too late to make use of a slot with a 1030Z CTA). However, another airline out there that wants a slot with a 1030Z CTA, and would gladly give up an 1100Z slot to get it. We know this because the ground delay program software detects that another airline has a flight with an 1100Z CTA and has set its ERTA to 1030Z or earlier.

In this case the steps in the process work as follows:

Airline ABC sends a message saying:

I am willing cancel Flight 1 and give up my 1030Z CTA *only if*

a. I can get a slot at 1100Z and put Flight 2 into it
b. Moving Flight 2 frees up a CTA at 1200Z into which I will put Flight 3
c. Moving Flight 3 frees up a CTA at 1300Z into which I will put Flight 4.

The ground delay program software looks for a flight to form the necessary bridge and when it finds the one mentioned above, it approves the entire transaction. (If it could *not* find a suitable flight to form the bridge, it would have rejected the entire transaction and Airline ABC would not have been obligated to cancel Flight 1.)

Note that it is not the setting of the ERTA that kicks off the Slot-Credit Substitution process. (ERTAs are always set as a matter of course, by default set as the originally scheduled or filed arrival time for each flight.) Rather, it is the airline's request for a Slot-Credit Substitution transaction that causes the ground delay program software to search for a suitable bridge.

Overall, then, this process has truly distributed the work by changing the locus of control and the parameters of control of certain decisions. The ATCSCC traffic manager's task is to assign the parameters defining the GDP. The ATC Coordinators for each NAS User can then, subject to the constraints imposed by the established GDP, make decisions about which flights to assign to which slots and can even, by setting ERTAs, offer to swap arrival slots with other NAS Users.

Flight	Scheduled Arrival Time	ERTA(Z)	CTA(Z)	Delay (mins)
1	1000	1000	1030	30
2	1100	1100	1200	60
3	1200	1200	1300	60
4	1300	1300	1400	60

FIGURE 29.4. Times associated with flights from LGA-ORD with departure delays assigned by a GDP.

Adapting to deal with uncertainty. The GDP process as described thus far basically assumes that the NAS is a deterministic process. If this assumption was true, and if the decision makers had perfect information, then the process would run in a flawless manner, with the right number of flights arriving at the constrained airport at the right time.

However, there are numerous sources of uncertainty in the NAS. The duration and extent of weather constraints are not totally predictable; the estimated time required to clear an aircraft with a blown tire off a runway may be wrong; a flight may miss its assigned departure slot (calculated by working backward from its assigned arrival time and its estimated air time); or a flight may take more or less time than predicted to arrive once airborne. To deal with these uncertainties, certain forms of adaptation have been built into the process.

First, ATCSCC can override the slot allocation for any individual flight, and assign it some other earlier time because of an extenuating circumstance. (In the parlance of the field, such a flight is referred to as a "white hat.") Second, if there are a number of slots before the end of the GDP that have not been filled with active flights, ATCSCC can run a "compression" algorithm that moves later flights up into those slots (subject to the constraints imposed by the ERTAs for those flights and the rule that a flight cannot be moved up to a slot that would require it to depart within the next 30 minutes). Third, ATCSCC can revise the GDP itself because the original predictions or resultant decisions have turned out to be too inaccurate, setting new parameters (such as a new AAR or a new end time), and causing a major shift in arrival slot assignments.

A fourth method for dealing with uncertainty is to pad the original estimate for the AAR. The major reason for doing this is to deal with potential "pop-up" flights. As described earlier, slots are allocated based on the available schedule for arrivals at an airport. However, many general aviation aircraft do not publish a schedule indicating where and when they plan to fly on a given day. They fly on an as-needed basis for their owners. Nevertheless, they still need to be accommodated by the system. To deal with such flights, when setting the AAR using FSM, the ATCSCC traffic manager can indicate the number of pop-ups to plan for, thus leaving some open slots for those flights. When such a flight files its flight plan, it is then assigned the average delay that was assigned to those flights that are now, under the GDP, expected to arrive in the same 15-minute period.

Administrative controls. In addition to these real-time decisions, the overall process is monitored for problems. One such potential problem is the quality of the data submitted by a NAS User. To deal with this issue, three metrics are monitored:

- Time-out cancels. A time-out cancel is a flight that is expected to operate (one that was scheduled, but for which a cancellation message was never submitted), but either never operates, or operates well after its assigned departure slot.

- Cancelled-but-flew flights. A cancelled-but-flew flight is a flight that the participant cancels but that ends up operating.
- Undeclared flights. An undeclared flight is a flight belonging to a NAS User that normally submits a schedule of its expected flights for each day, and that operates without prior notice to the system.

NAS Users must demonstrate acceptable performance in terms of these metrics in order to continue to participate in using the slot-swapping functions provided by GDPs. This helps to ensure sufficient data integrity for the overall process to function as planned.

A second area requiring administrative oversight is a concern over gaming. Suppose, for example, that SFO (San Francisco) has a GDP in effect because of reduced visibility at that airport. A flight that wants to fly to SFO should take a delay based on this GDP. However, if it files for OAK (Oakland, an airport very close to SFO) instead, and then requests an amendment to land instead at SFO once airborne, it has violated the spirit of the GDP program. ATCSCC monitors for such practices, and takes administrative actions to prevent them in the future.

Evaluation of the Implemented Solution

There are two components of the benefits from the Enhanced Ground Delay Program which has been described above in terms of its major features. The first is the benefit to the NAS Users derived from swapping flights in order to reduce delays for their higher-priority flights. These savings are clearly very substantial, but would be very difficult and time consuming to estimate. As a result, to date no study has been conducted to put a dollar figure on that class of benefits.

The second benefit arises from the use of the compression process to fill in unused arrival slots, therefore reducing overall delays. Figure 29.5 shows the frequency of use of the Enhanced GDP process since the introduction of the compression process in 1999. Note that, over the past two years, it has not been uncommon to run as many as 100 GDPs per month. Table 29.1 then provides an estimate of the minutes and dollars saved by the compression process alone, using a conservative average cost per minute of operation for a flight of $42. These estimates suggest that the use of the compression process alone reduced delay by 28,188,715 minutes from March 18, 1999 to June 30, 2005, thus saving over 1 billion dollars.

Case Study A—Conclusions

The Enhanced GDP illustrates a deliberate effort to design a distributed work system. The design strategy is to have one party (ATCSCC as a neutral referee) set a high-level constraint (the airport arrival rate), and to then let the other involved parties (the NAS Users) find solutions that meet their business needs as effectively as possible subject to that constraint (by swapping slots among flights to favor the highest-priority flights). Taking this approach, knowledge and data remain distributed, matching the locus of control for the various decisions that need to be made.

This distribution also serves to reduce the cognitive complexity of the task confronting each individual. The ATCSCC traffic manager doesn't need to know anything about User priorities, and the ATC Coordinator for a NAS User doesn't need to become an expert at setting airport arrival rates. Finally, because exceptions are necessary to override poor decisions when they are proposed by someone due to a slip or due to insufficient access to the relevant knowledge or data, there must be a mechanism to support interactions among the different participants in order to exchange the necessary information and modify those decisions.

To make this process work, however, certain other requirements arise during its design:

- To make it politically acceptable, non-participants (NAS Users who do not choose to participate in slot swapping, and so on) must not be penalized. Such non-participants are initially assigned slots just like everyone else, but their flights are

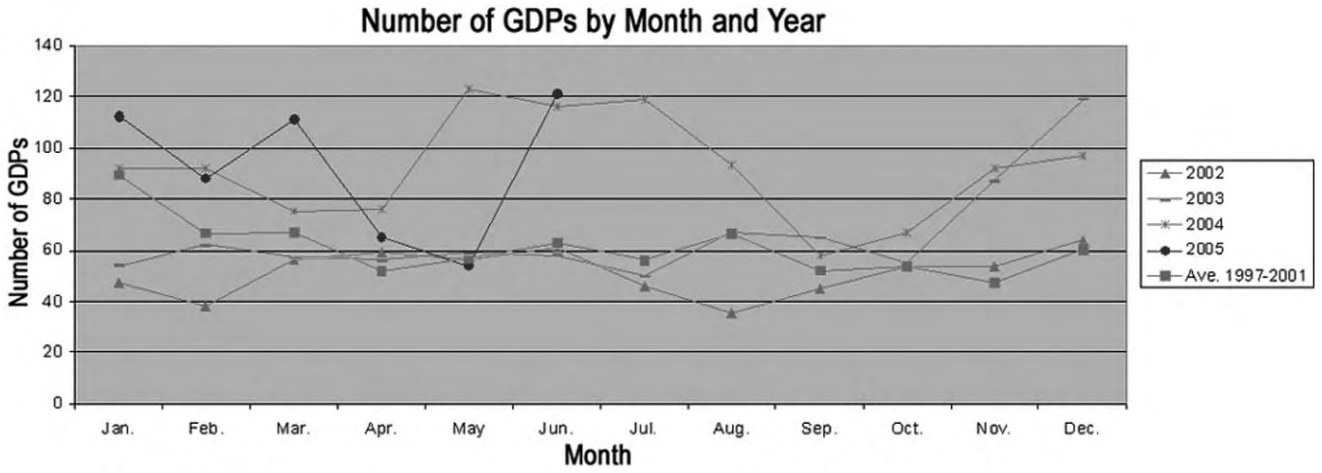


FIGURE 29.5. Frequency of use of the enhanced GDP by year.

TABLE 29.1. Time and Money Savings from the use of the Compression Process in GDPs since 1999

- Data Included in Analysis:
  - All compression from Mar. 18, 1999 – June 30, 2005
  - 3,573,796 flights included
  - 28,188,715 total minutes reduced
  - $1,183,926,030 (@$42/min) dollars saved
- Departure Delay Performance
  - Average reduction: 7.7 minutes
- Percentiles:
  - Reductions greater than 15 minutes: 12.3% (440,889 flights)
  - Reductions greater than 30 minutes: 6.1% (218,886 flights)
  - Reductions greater than 45 minutes: 3.8% (137,424 flights)
  - Reductions greater than 60 minutes: 2.6% (93,558 flights)

never moved to a later slot as part of any swapping. (They may actually benefit from compression, however, if they are moved up to an earlier slot that is empty.)

- There must be an incentive for information sharing.
- DSTs must be available to provide:
  - digital communication;
  - simulation of the predicted impact of alternative decisions (what-if capabilities);
  - algorithmic support for identifying, evaluating, and implementing complex decisions involving large numbers of flights (swapping);
  - information displays that allow the user to understand the situation as relevant to that user's responsibilities, and to effectively evaluate the solution recommended by the technology;
  - control by the user to override the DSTs recommendations in cases where the technology exhibits brittleness.
- Organizational changes in roles and responsibilities (e.g., new specialists at ATCSCC with the knowledge and skills to develop GDPs; ATC Coordinators for the NAS Users to look at tradeoffs in delaying alternative flights).
- The overall procedures must support adaptation, so that decisions can be modified as the scenario evolves.

In summary, this case study serves to illustrate the importance of looking at design from a broad systems perspective when dealing with a distributed work environment, considering not just the detailed design issues necessary to implement a given DST, but also considering the alternative strategies for distributing the work and the organizational changes required to match people with certain skills and knowledge with corresponding responsibilities, and to make the system resilient in the face of slips, mistakes, and unanticipated events.

Case Study B—Human Interactions with Autonomous Mobile Robotic Systems

A rapidly emerging technical area is that of autonomous mobile robotic systems. An autonomous mobile robotic system can perform complex tasks for extended periods with no human intervention, and thus represents a large step beyond remote control. But no system is truly autonomous—interaction with its human owners and operators is inevitable. This case study illustrates the issues and needed approaches to provide an effective and responsible relationship between a human supervisor and a team of autonomous mobile robotic systems.

Remote control of mobile robotic vehicle systems has become commonplace over the past two decades. Remote control depends on a human operator to visualize the situation around the remote vehicle and to provide control inputs to maneuver the vehicle in space and to manage its sensors and manipulators to achieve the desired mission. Examples of remote control of robotic vehicles include deep ocean exploration and repairs, bomb disposal operations, and handling of very hazardous materials such as radioactive waste.

While remote control has been successfully used for many situations in which it would be highly undesirable to attempt to send a human, the requirement to maintain continuous information and control flow between the vehicle and the human operator has been limiting. Over the past decade, significant investment has been made in research and development of autonomous mobile robotic systems with promising results for land, air, and undersea vehicles.

The study of human interaction with remotely operated and autonomous systems has a long history. The classic work was by Sheridan and Verplank (1978) because of their study of issues for remote operations of undersea robotic vehicles, and has been recently updated in Parasuraman et al. (2000). This work defines ten levels of autonomy, ranging from fully manual to fully autonomous. Most significantly, the levels are distinguished by defining the nature of the interactions between the human and the machine.

The highest six levels are particularly relevant to this case study:

10. The computer decides everything, acts autonomously, ignoring the human
9. (The computer) informs the human only if it, the computer decides to
8 (The computer) informs the human only if asked, or
7. (The computer) executes automatically, then necessarily informs the human, and
6. (The computer) allows the human a restricted time to veto before automatic execution, or
5. executes that suggestion if the human approves, or . . . (p. 933)

Thus, like many other issues in human-computer interaction, creating a decision-support system for control of a team of autonomous entities requires careful attention to the cognitive characteristics of humans.

The Application Area

During the mid-1990s, as a result of the earlier successful demonstrations of intelligent systems technology within the military, key technology decision makers with the defense industry research and development community recognized that it was possible to create an unmanned combat air vehicle.

The unmanned combat air vehicle, or UCAV, was envisioned as a platform for performing very dangerous missions or, alternatively, very routine missions. An example of a dangerous mission

for modern combat aircraft is *suppression of enemy air defenses*, or SEAD. In the SEAD mission, a team of UCAVs would be assigned the role of finding and attacking enemy surface-to-air missile sites, so that the airspace over the battle area would be safe for manned aircraft.

The development of unmanned reconnaissance aircraft has made steady progress, resulting in the deployment of the Predator and Global Hawk unmanned aircraft systems. While the Predator is still a remote controlled system, flown by a human pilot at a ground station, the Global Hawk is a true autonomous aircraft. These successes raised confidence that the UCAV concept was feasible as well.

The demands of combat aviation are, however, quite different from reconnaissance and surveillance. Reconnaissance missions normally are single-aircraft missions, planned so that the aircraft avoids potential threats. The typical mission also has well-defined locations where information is sought, and success depends on collecting the information without being detected.

Combat aviation, in contrast, is conducted in an environment of uncertainty and hostility. The enemy actively seeks to deny information about his location and his intent by blocking and disrupting sensors and communications, and by using decoys. The enemy is also using his defensive weapons to find and destroy his opponent's forces. In this environment, a team of combat aircraft must operate successfully with degraded information and limited contact with human supervisors. The team must be able to find and attack the enemy while avoiding damage to noncombatants. At the same time, the team must be able to recognize and remediate system degradations and failures without assistance.

For an unmanned combat aircraft, the difficulty of this environment is higher still. The human supervisor no longer has the direct sensory perceptions available to a pilot flying on the aircraft. His entire understanding of the situation depends on limited sensor data, and his ability to control the responses to the situation is hindered by time lags and dropouts of data due to remote data transmissions. To be successful, the UCAV concept needs to be robust and flexible in the face of a decidedly hostile environment.

The Need for Decision Aiding

While the current generation of unmanned reconnaissance and surveillance aircraft has been successful, this success has largely been dependent on the skill and direct activity of their human operators. Several humans are required to operate the Predator, which is actually remotely flown and not an autonomous system. Similarly, Global Hawk requires a team of humans that continuously monitors a single aircraft throughout its mission, as well as a significant mission-planning effort prior to launch.

If unmanned combat air vehicles are to be useful, a significant reduction in human-operator effort is required. Modern manned aircraft, such as the F-22 Raptor and the newly designed F/A-35 Joint Strike Fighter, are flown by a single human pilot. While training of a military combat pilot is expensive and takes considerable time, UCAVs will not be accepted as an adjunct to manned aircraft unless the mission can be reliably performed and both (a) the cost of the unmanned aircraft is less than a manned aircraft and (b) the cost of the human operators needed for one unmanned air vehicle is significantly less than the cost of training and supporting a pilot for a manned aircraft.

As a result, the design goal for unmanned combat aircraft is typically stated as a single operator controlling a flight of four or more UCAVs in a combat setting. Given the issues that confront a human operator of an unmanned combat aircraft mission, researchers (Elmore, Dunlap, & Campbell, 2001; Geddes & Lee, 1998) have recognized that decision aiding is needed across a broad front:

- Situation awareness. The human operator must understand the hostile external environment, the offensive and defensive postures of all of the aircraft under his control, and the internal status of all of the aircraft systems for all of the aircraft under his control. This is a massive amount of information to digest, with the potential for sudden and significant changes at any moment.
- Planning and resource commitment. The UCAVs are assigned a mission, and the human operator is responsible for managing the frequently changing tactical behaviors needed to meet the mission goals. Generating alternative short-term tactical plans, evaluating the alternatives, selecting the most effective plans and committing the aircraft resources to the plans is a continuous requirement.
- Coordination and Acting. As plans are generated and selected, the human operator must also ensure that the plans are coordinated across the flight, free of conflicts, and performed as expected. At any point in time, multiple plans across subsets of the UCAVs in the flight will be active and will require monitoring.
- Understanding and responding to irregularities. Combat aircraft, sensors, and weapons are extremely complex, and as a result, behavior anomalies are common. In addition, the enemy seeks to exploit vulnerabilities in systems to create irregularities. Recognizing abnormal situations and responding in a timely way is a frequent requirement in combat.

In meeting these cognitive needs of the human operator, the designer must be careful not to create a "black box" automation that blocks human involvement (Wiener & Curry, 1980). More recently, a related but distinctly different challenge has emerged—that of "point intelligence" that is not as robust and flexible as it appears (Geddes, 1997b; Hammer & Singletary, 2004). Point intelligence is designed to solve a particular narrow problem, such as assessment or planning, but does not have broad competency. As Hammer and Singletary (2004) pointed out, this leads to behavioral breakdowns between the human and computer.

The Design Solution

From 2002 through 2005, the U.S. Air Force Research Laboratories sponsored the development and evaluation of an approach to supporting human management of autonomous unmanned combat air vehicles (Atkinson, 2003, 2005). This system, known as AUCTIONEER, combines a cognitive model of decision making with a market auctioning system. The result

is a system that is capable of seamless operation across Levels 5–9 of Sheridan's classification of levels of autonomy.

As discussed by Atkinson, the functional organization of a complex vehicle, such as a combat aircraft, normally has a number of layers, as shown in Fig. 29.6. The lowest layer acts to control the physical parts of the aircraft, such as its flight control surfaces, engine control surfaces and fuel metering, sensors, and weapons. The next layer contains functions that integrate the behaviors of the lowest layer into more complex behaviors to perform tasks. For example, the Navigation System may act to receive position and velocity data from the sensors and direct the flight controls and engines to new settings in order to fly the aircraft toward a desired waypoint in space. This second layer contains many functions that are typically implemented as conventional automation, with limited provisions for human interactions. Even so, a modern manned combat aircraft cockpit may have over 350 controls for a pilot to manipulate to configure his vehicle, sensors, and weapons systems.

The third layer in Fig. 29.6 provides the overall mission organization of the aircraft's behaviors, so that the intended utility of the system can be achieved. For manned aircraft, this layer is mostly performed by a human pilot. The typical training time for a carefully selected person to reach threshold proficiency as a combat aircraft pilot is two years, and expertise normally requires 10 years of continued training and practice.

The fourth layer, the coordination layer, is often performed by humans who are distantly located as part of a command-and-control structure. With the present generation of manned aircraft and command and control systems, little "computer-to-computer" coordination is available. Most coordination is performed human-to-human.

If unmanned vehicles are to meet their economic goals by using significantly fewer human operators, the mission layer and the coordination layer must both be substantially more autonomous. Yet, because an unmanned combat system must also be able to apply lethal weapons, uninterrupted autonomy for any significant periods of the mission may compromise safety and human responsibility. As a result, operations at Level 10 of Sheridan's hierarchy are strongly undesirable. However, seamless operation across Levels 5–9 of Sheridan's model of autonomy is strongly required.

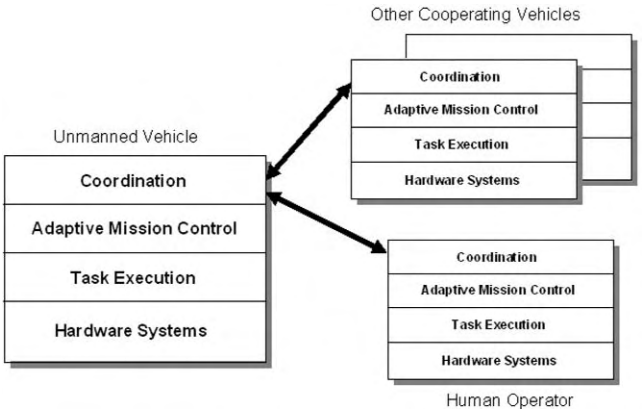


FIGURE 29.6. Behaviors of complex systems are produced in layers.

The AUCTIONEER system is based on the architecture from Elmore et al. (2001), which has evolved directly from the original Pilot's Associate architecture (Lizza & Banks, 1991; Rouse, Geddes, & Hammer, 1990) of the early 1990s. This architecture uses a distributed system of cognitive engines, providing full situation assessment, planning, coordinating, and acting for each entity in the system, both human and UCAV.

To meet the need for situation awareness, the AUCTIONEER system provides for a distributed model of beliefs that is shared across the unmanned members of the flight and the human operator. The belief model, embodied as a concept graph, allows each aircraft to combine information from other aircraft with its own and with that provided by other sources, as well as input from the human. The belief model also provides for active monitoring for specific events that are coupled to the ongoing and planned tasks. When a task is initiated, specific monitors are activated in the belief model. When monitored events are detected, existing tasks may be modified or new tasks activated by the planning functions.

Distributed planning for AUCTIONEER is performed using a market-based auctioning system that has no centralized auction manager service. Each member of the flight can initiate an auction for a task that it determines needs to be performed (because of a monitored event), and any members of the flight, including the auction initiator, can bid on the task. More than one auction can be proceeding simultaneously. The auction mechanism does not assume that all of the members can receive the auction announcement, or that all of the bids can be received in return because of communications dropouts.

The auctioning system uses a carefully structured representation of the tasks that may need to be performed during the mission. The structure is a "Plan Goal Graph" consisting of a directed acyclic graph with alternating plan and goal nodes. The Plan Goal Graph defines the network of causal relationships between activities described by plan nodes and goal nodes that reflect what each activity is expected to accomplish. The Plan Goal Graph has the additional capability to be used as a model of the intentions of groups of active entities (Geddes, 1997a), and is used in AUCTIONEER to both create the tasks to be auctioned and to interpret the actions of the members of the flight and the human operator as the tasks are performed.

Similar studies by Stenz, Dias, and others at Carnegie Mellon (Stenz & Dias, 1999; Zlot & Stenz, 2003) have shown that market-based auctioning on tree-structured tasks has many computational advantages over either the auctioning of unstructured primitive tasks or the use of algorithms that attempt optimal task allocations. However, successful use of an auctioning system must provide mechanisms for the human operator to understand the processes and outcomes of the auctions and to intervene in auctioned tasks without disrupting the benefits of using market-based auctions as a means for resource allocation.

For example, in Fig. 29.7, the user can see the approved paths and proposed paths of vehicles in the flight; review past, future, and proposed tasking for each target or for each vehicle; and edit proposed and active tasks if desired.

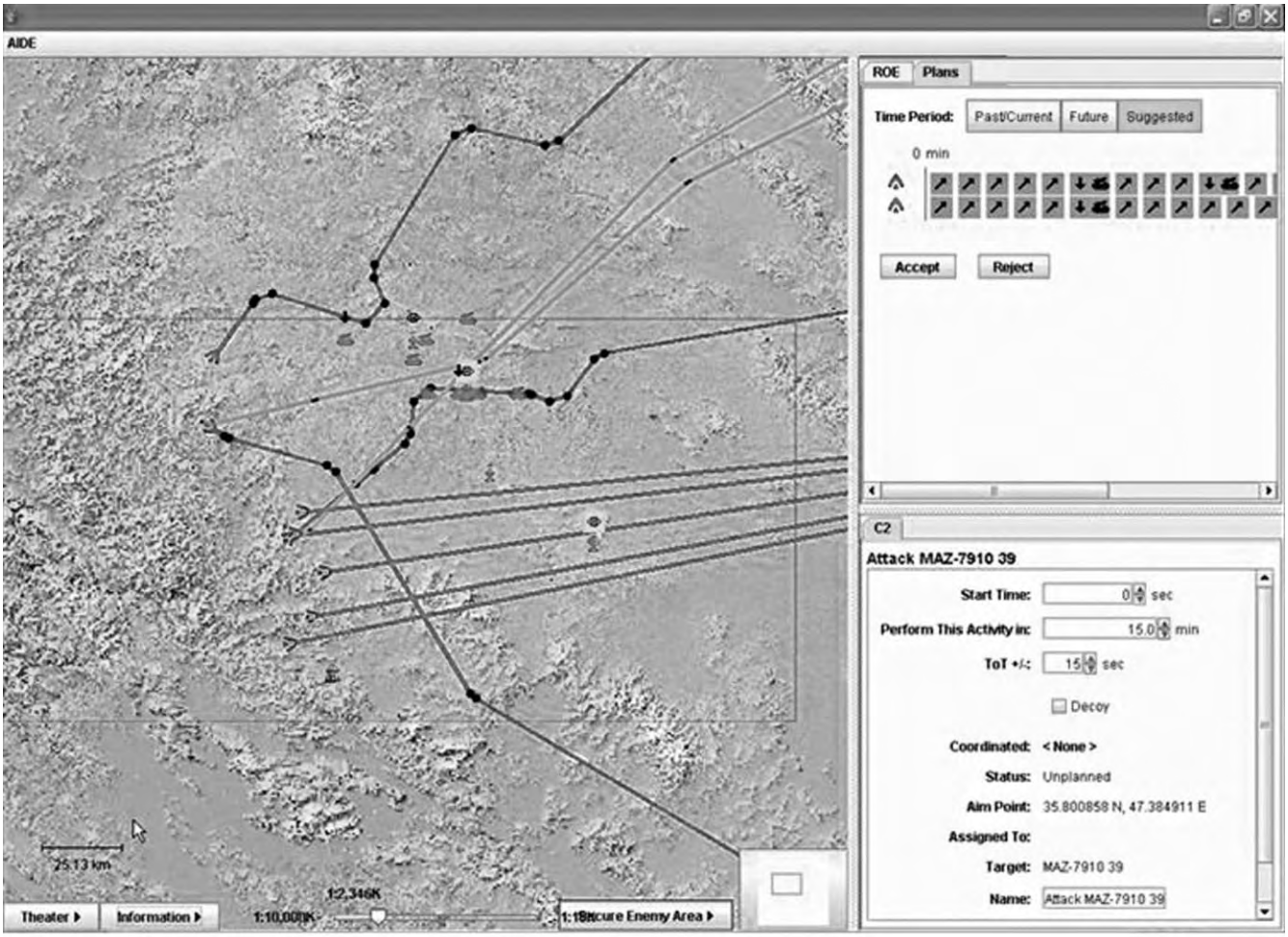


FIGURE 29.7. The set of UCAVs have begun execution of one coordinated attack while proposing a second coordinated attack against a different target complex. The routes in the lighter grey (magenta in true color) are for the proposed attack, while the darker routes are the currently accepted paths. The group of straight paths in the lower right of center are for vehicles that have not yet won any of the auctioned tasks.

Results

The AUCTIONEER study is particularly useful as a case study of the effects of opening the "black box" of automation because the system was studied with both passive and active operator involvement.

Performance. In the first study results (Atkinson, 2003), the auction processes were run without human intervention. This study established the basis for measurements of effectiveness for the auction environment as a fully autonomous process. This first study found that the auction process was an effective method for allocating resources in terms of mission accomplishment. More targets were attacked in a shorter time with fewer abandoned tasks than what could be achieved by unaided human control of the vehicles. A second concern was timeliness, because the SEAD mission requires that targets are attacked or suppressed before they can disrupt the friendly forces. The auction process also resulted in shorter average timelines to attack targets.

The most significant findings dealt with the stability of tasking. Because the winner of an auction might later find a better target before finishing the current task, winners could become auctioneers and "sell" their current tasks to another. Atkinson found similar results to Zlot and Stenz: that this reselling process did not result in churning of tasks and in fact led to improved outcomes.

Acceptance. The second study (Atkinson, 2005) added the human operator back into the mission, with the ability to intervene in the auction process as described above. In this study, the overall findings showed that the operator's understanding and effectiveness were limited by the lack of insight into the auction process. Even though the underlying decision-support model used appropriate human cognitive constructs, the system was still a black box to the human because it did not reveal

its internal model at the user interface. Atkinson and her colleagues provided a recommended set of features to open the black box of the auction process.

Communication of beliefs. The operator should be able to view the beliefs held by each member of the flight, and act to modify those beliefs when required. This provides assurance that the members of the flight are acting with a common assessment of the situation and are exploiting all available information.

Communications of intentions in the context of a bid. If the only information available to the operator about an offered bid for a task is its total cost, the operator loses important information about the context of the bid. By allowing the operator to view the chain of intentions that led to the bid, as represented in the Plan Goal Graph, the operator would be able to understand how the bid was composed, and to be assured that the bid was proper.

Communication of direct commands. Operator commands need to be recognized as distinct from task-auction requests. This would allow the operator to act with assurance that his commanded action would be carried out by the aircraft and resource that was directed, rather than incurring an auction response of uncertain outcome. Any additional task reallocations that result from the operator's commands may be automatically handled as auctions.

The findings in this case study underscore two important issues in designing intelligent decision-support systems: (a) the importance of avoiding "black box" automation for high-stakes environments, and (b) the advantages of designing and building decision-support systems that are based on models of human cognitive processes.

While it may seem easier to design and implement approaches that are mathematically elegant, this can result in an automation "black box" that undermines human understanding. Similarly, efforts to reduce the scope of the decision support to solve just one part of the operator's cognitive problem may result in point intelligence that can also degrade performance.

Humans learn to interact with other humans as part of the development of social skills for effective behaviors within their cultures. In cultures based on trust and mutual accomplishment, the ability to communicate about beliefs, goals, and intentions is an important skill. This case study suggests that creating automated behavioral systems that can use explicit models of beliefs, goals, and intentions as the basis for communication with humans is an effective way to open the black box.

Case Study C—Interactive Critiquing as a Form of Decision Support

This case study emphasizes the challenge for interaction designers in dealing with the use of decision-support tools that have the potential for brittle performance due to known or unknown limitations. While there are no perfect solutions to this problem, there are a number of approaches to help reduce its impact.

This case study looks at a tool that incorporates three complementary approaches to the design of an expert system in order to improve overall performance, while reducing the potential impact of brittle performance by the expert system. The first approach to deal with the impact of brittle performance by an expert system is to design a role that encourages the user to be fully engaged in the problem solving, and to apply his knowledge independently without being first influenced by the software. In this case study, the approach explored to achieve this is the use of the computer as a critic, rather than as the initial problem-solver (Fischer, Lemke, Mastaglio, & Morch, 1991; P. Miller, 1986; Silverman, 1992; Tianfield & Wang, 2004). Thus, instead of asking the person to critique the performance of the software, the computer is assigned the role of watching over the person's shoulder. Note that this is more accurately described as having the design team try to anticipate all of the scenarios that can arise, and then, for all of those scenarios, trying to incorporate the knowledge necessary to detect possible slips or mistakes on the part user and to provide alerts and assistance in recovering (Pritchett, Vandor, & Edwards, 2002; Wogalter & Mayhorn, 2005).

The second approach to reduce sensitivity to brittleness in the computer's performance is to incorporate metaknowledge into the expert system that can help it to recognize situations in which it may not be fully competent. By doing so, the software may be able to alert the person to be especially careful because the computer recognizes that this is an unusual or difficult case.

The third approach to deal with brittleness is to develop problem-solving strategies that reduce susceptibility to the impact of slips or mistakes. In the software discussed in this case study this is accomplished by incorporating a problem-solving strategy that includes the collection of converging evidence using multiple independent problem-solving strategies and sources of data to arrive at a conclusion.

What follows, then, is a discussion of a specific expert system that incorporates these three strategies for dealing with brittleness. Empirical testing of the system suggests that this approach can significantly enhance performance, even in cases where the software is not fully competent (P. J. Smith & Rudmann, 2005).

The Application Area

The specific problem area considered in this case study is the design of a decision-support tool to assist blood bankers in the identification of compatible blood for a transfusion. One of the difficult tasks that blood bankers must complete as part of this process is the determination of whether the patient has any alloantibodies present in his blood serum, and if so, what particular antibodies are present.

Diagnosis as a generic task. Abstractly, this is a classic example of the generic task of abduction or diagnosis (J. Josephson & S. Josephson, 1994; Psillos, 2002). It involves deciding which tests to run (what data to collect), collecting and interpreting those data, and forming hypotheses, as well as deciding what overall combination of problem-solving strategies to

employ. Characteristics of this generic task that make it difficult for people include:

- The occurrence of multiple-solution problems, where more than one "primitive" problem is present at the same time;
- The occurrence of cases where two or more "primitive" problems are present, and where one problem masks the data indicative of the presence of the other;
- The existence of "noisy" data;
- The existence of a large "data space" where data must be collected sequentially, so that the person must decide what data to collect next and when to stop;
- The presence of time stress, in which an answer must be determined quickly (Elstein, Shulman, & Sprafka, 1978; P. J. Smith et al., 1998).

In this application, the "primitive" problems are the individual antibodies that may be present in the patient's blood serum. Time stress can arise when the patient is in an emergency and needs a transfusion quickly.

Sample problem. To illustrate the nature of this diagnosis task, consider the following partial description of an interaction with the decision-support tool AIDA (the Antibody IDentification Assistant) that is the focus of this case study (Guerlain et al., 1999). Initially, the medical technologist needs to determine whether the patient is type A, B, AB, or O, and whether the patient is Rh positive. Then the technologist determines whether the patient shows evidence of any autoantibodies or alloantibodies. As part of this determination, displays like the one shown in Fig. 29.8 are provided by AIDA.

To make visual scanning easier on this data display, some of the rows have been highlighted by the technologist in yellow (shown in light gray in this black-and-white version). In addition, to reduce memory load, the technologist has marked a number of intermediate conclusions, indicating that the patient could have antibodies against the C, E, M, and N antigens (indicated by the technologist in orange in the actual system, but left white in Fig. 29.8), that the f, V, Cw, Lua, Kpa, and Jsa antibodies are unlikely (marked in blue in the actual system, but light gray in Fig. 29.8), and that the other antibodies shown as the labels

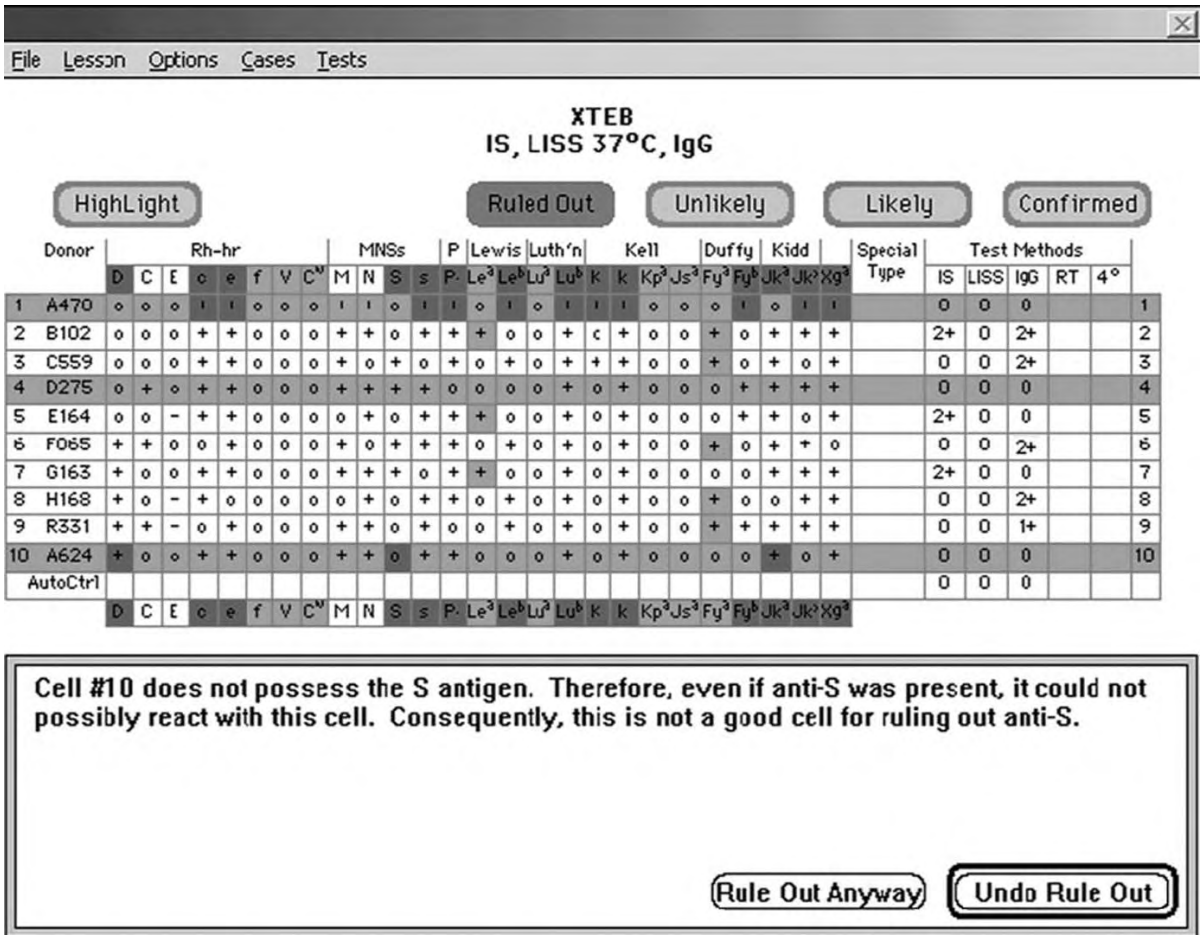


FIGURE 29.8. Full test panel with intermediate results marked using color coded markers provided by AIDA (shown here in black and white). In this case, AIDA has interrupted the user to suggest that an error may have been made in ruling out anti-S.

across the top row can be ruled out (marked in green in the actual system but shown in darker gray on this black-and-white version.) These color-coded intermediate answers are transferred to all other data displays to reduce the memory load for the technologist. Figure 29.8 also provides an example of a critique that AIDA has provided in response to an error made by the technologist in marking anti-S as ruled out. This critique was generated by the expert model (rule-based system) underlying AIDA, which monitors all of the test selections and markings made by the technologist as he solves the case using the interface provided by AIDA.

Thus, this figure serves to illustrate that:

- The technologist is responsible for completing the analysis, deciding which tests to run and which intermediate and final conclusions to reach, and is thus very engaged in the task;
- The computer provides an interface that makes it easy for the technologist to select the tests to run and to view the resultant data (using the highlighting to make visual scanning easier), as well as to remember intermediate conclusions (using color-coded markings to indicate these conclusions);
- Although the primary motivation for the technologist in marking the data forms is to make his task easier, when he does so the computer is provided with a great deal of data regarding what the user is thinking, and can use those data to make inferences based on its expert model about when to provide a critique cautioning the user that he may have made a slip or mistake.

The need for decision aiding. Initial laboratory and field studies indicated that the task of determining the alloantibodies present in a patient's blood is a difficult one for many technologists. A variety of causes of errors were observed, including slips, perceptual distortions, incorrect or incomplete knowledge (Guerlain et al., 1999), and cognitive biases (Brachman & Levesque, 2004; Bradfield & Wells, 2005; Fraser et al., 1992; Gilovich et al., 2002; Haselton et al., 2005; Kahneman & Tversky, 2000; Plous, 1993; Poulton, 1989).

The Design Solution

It was the judgment of the design team that this task was sufficiently complex that, given the available development resources, it was unlikely that all of the possible scenarios could be anticipated and dealt with by the design team. It was also noted that, for this task, the cost of an erroneous diagnosis was potentially high. Thus it was decided that, instead of automating the task, a decision-support system should be developed that would keep the person very much engaged in the task, and provide other safety nets to reduce the chances of error for those cases in which the computer's knowledge was incomplete. This conclusion was reinforced by a preliminary study regarding the impact of role on diagnostic accuracy in antibody identification. Guerlain et al. (1996) showed that, when the user was asked to critique the computer's performance, instead of having the computer critique the user's performance, the final answer was wrong 31% more often in cases where the computer's knowledge was incomplete when, in that case, the person was assigned the role of critic rather than having the computer critique the person.

Critiquing—additional design considerations. The literature provided additional guidance in deciding whether and how to develop this software to play the role of a critic. For example, P. Miller (1986) developed a prototype system called ATTENDING that focused on anesthesiology. Based on studies of its use, Miller suggested that critiquing systems are most effective in applications where the user has a task that is frequently performed, but that requires the person to remember and apply a great deal of information in order to complete a case. Miller's conclusion was that, on such tasks, the person is more susceptible to slips and mistakes and would therefore benefit significantly from the decision-support system.

A second consideration was how intrusive the interactions with the knowledge-based system would be for the user. A number of researchers have suggested that an interface that requires that the user spend significant time entering data and informing the computer about what he has concluded is too cumbersome, and will therefore be unlikely to be adopted in actual practice (Berner, Brooks, R. Miller, Masarie, & Jackson, 1989; Harris & Owens, 1986; Miller, 1986; Shortliffe, 1990; Tianfield & Wang, 2004).

A third consideration was concern over the potential for complacency if the person played the role of critic, letting the computer complete an initial assessment and then having the person decide whether to accept this assessment. Parasuraman and Riley (1997) have shown that, in such a role, there is a risk that the person will become overreliant on the computer, and will not adequately apply his knowledge in completing the critique (Metzger & Parasuraman, 2005). (Note, however, that a person could become overreliant even with the roles reversed, as the person might start to get careless and assume the computer will always catch his slips. Administrative controls, based on regular monitoring of the person's performance, might help reduce such complacency, but this is an as-of-yet unexplored aspect of the use of critiquing systems.)

A final consideration was the timeliness of critiques. Waiting for the user to reach a final answer before providing a critique, as has been the case for many critiquing systems, is potentially inefficient and objectionable, as the user may have invested considerable time and effort in arriving at a mistaken answer that the computer could have headed off earlier in the person's problem-solving process. Furthermore, if the critique is given well after the user has made a slip or mistake, it may be difficult for him to remember exactly why he arrived at that intermediate conclusion (thus making it harder to decide whether to accept the computer's critique). This consideration therefore suggests the need for an interface that provides the computer with data about the user's intermediate conclusions rather than just the user's final answer, so that critiquing can be more interactive.

Based on these considerations, AIDA was developed as a critiquing system that supported the user as he completed the antibody identification task. In order to provide immediate, context-sensitive critiques, an interface was developed that encouraged the user to mark intermediate conclusions on the screen.

As suggested earlier, these markings in fact reduced the perceptual and memory loads for the user, thus implicitly encouraging this form of communication and allowing the computer to detect and respond immediately with context sensitive critiques to potential slips and errors made by the person.

Complementary strategies to reduce susceptibility to brittleness. The considerations outlined above focused on how to keep the person engaged in the diagnosis task, and how to avoid triggering the cognitive biases that can arise if the computer suggests an answer before the person has explored the data himself (Larson & C. Hayes, 2005; P. J. Smith, McCoy, & Layton, 1997). AIDA also incorporated two additional design strategies to reduce the potential for error. One was the incorporation of metaknowledge into the knowledge-based system. The other was to include a problem-solving strategy that was robust even in the face of slips or mistakes by either the person or the computer (e.g., the design team).

Metaknowledge was included to help the computer identify cases in which its knowledge might be incomplete. Such rules were developed by identifying the potential weak points in the computer's problem-solving process. An example was the use of thresholding by AIDA when applying rules such as:

If a test cell's reactions are 0 on a test panel for all three of the testing conditions (IS, LISS and IgG) as they are for the first test cell (Donor A478) shown in Figure 29.8, and if e is present (shown by a + in the column labeled e in the row corresponding to the first test cell) on that test cell but E is not (shown by a 0 in the column labeled E in the row corresponding to the first test cell), then mark e as ruled out.

This heuristic usually produces the correct inference. Since it does not directly reason with some form of probabilistic reasoning (Bochman, 2004; Pearl, 1988; Shafer, 1996; Shafer & Pearl, 1990), however, it can lead to ruling out an antibody that is actually present in the patient's serum with no consideration of the level of uncertainty. This is most likely to happen when the reaction strengths are weak. Thus, AIDA was provided with a rule that monitored for weak reactions on cells, and when this situation was detected and the user was observed trying to complete rule-outs without first enhancing the reactions with some alternative test phase, the system cautioned the user. In this way, AIDA put the user on alert to be especially careful in applying his normal strategies and rules.

A second strategy incorporated into AIDA as protection against brittleness was to monitor for the collection and consideration of converging evidence by the technologist. This problem-solving strategy was observed in use by one of the experts involved in the development of AIDA. She used this strategy to catch her own errors and those of the technologists working under her supervision. The strategy was based on the assumption that any one data source or line of reasoning could be susceptible to error, and that it is therefore wise to accept a conclusion only if independent types of data (test results that are not based on the same underlying data or process) and independent sets of heuristics or problem-solving strategies have been used to test that conclusion. Based on this expert strategy, AIDA monitored the user's problem-solving process to see whether such converging evidence had been collected prior to reaching a final answer. If not, AIDA cautioned the user and suggested types of converging evidence that could be collected.

TABLE 29.2. Percentage of Blood Bankers Arriving at the Wrong Final Answer on Four Test Cases

	Control Group	Treatment Group
Posttest Case 1	33%	0%
Posttest Case 2	50%	19%
Posttest Case 3	38%	0%
Posttest Case 4	63%	0%

Evaluation of the Implemented Solution

From the standpoint of human-computer interaction, the key question is how effectively this human-machine system performs using this cooperative problem-solving paradigm (Erkens, Andriessen, & Peters, 2003). To gain insights into this, an empirical study was conducted using AIDA.

This study of 37 practitioners at seven different hospitals found that those blood bankers using AIDA with its critiquing functions turned on made significantly fewer errors ($p < .01$) than those that used AIDA as a passive interface. Errors in the final diagnoses (answers) were reduced by 31–63%.

On those cases where AIDA was fully competent (Posttest Cases 1, 2, and 4), errors (% of technologists getting the wrong final answer) were reduced to 0% when the critiquing functions were on (Treatment Group). In the case where AIDA was not fully competent (Posttest Case 2), AIDA still helped reduce errors by 31%.

These empirical results support the potential value of interactive critiquing, supplemented with the metaknowledge and problem-solving strategies that were embedded in the software, as a design strategy for applications in which concerns regarding the potential brittleness of the technology are deemed significant.

CONCLUSION

One goal of this chapter has been to outline different conceptual approaches to support the design of DSSs, including the use of cognitive task analyses and work-domain analyses. A second goal has been to review relevant knowledge about the psychologies of both the designers and users of DSSs, identifying important design questions that need to be considered in the development of such software systems. Finally, the chapter emphasizes the need to take a broader systems perspective when deciding how to integrate DSSs into a work environment, as there are a number of other high-level design parameters (such as deciding how to distribute the work among a team of people and machines) that may offer alternative opportunities for effectively incorporating DSSs into the overall system.

References

Amalberti, R. (1999). Automation in aviation: A human factors perspective. In D. Garland, J. Wise, & V. Hopkins (Eds.), *Handbook of aviation* (pp. 173–192). Mahwah, NJ: Lawrence Erlbaum.

Anderson, J.R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J. R., Bothell, D., Byrne, N., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036–1060.

Andes, R. C., & Rouse, W .B. (1992). Specification of adaptive aiding systems. *Information and Decision Technology, 18*, 195–207.

Annett, J., Duncan, K. D., Stammers, R. B., & Gray, M. J. (1971). *Task analysis*. London: HMSO.

Annett, J., & Stanton, N. (2000). *Task analysis*. London: Taylor & Francis.

Atkinson, M. L. (2003). Contract nets for control of distributed agents in unmanned air vehicles. *Proceedings of 2nd AIAA Unmanned Unlimited Conference*, San Diego, CA.

Atkinson, M. L. (2005). Adding an operator into a task auction paradigm. *Proceedings of the AIAA InfoTech Conference, Workshop and Exhibit,* Arlington, VA, 29–35.

Baddeley, A. (1998). *Human memory: Theory and practice*. Boston: Allyn and Bacon.

Baddeley, A. (1999). *Essentials of human memory*. London: Taylor & Francis Group.

Baecker, R., Grudin, J. Buxton, W., & Greenberg, S. (1995). *Readings in groupware and computer-supported cooperative work: Assisting human-human collaboration*. San Francisco: Morgan Kaufmann.

Bainbridge, L. (1983). Ironies of automation. *Automatica, 19*, 775–779.

Bar-Yam, Y. (2003). *Dynamics of complex systems: Studies in nonlinearity*. Nashville, TN: Westview Press.

Bar-Yam, Y. (2005). *Making things work: Solving complex problems in a complex world*. Cambridge, MA: NECSI Knowledge Press.

Beach, L. R., & Connolly, T. (2005). *The psychology of decision making: People in* organizations (2nd ed.). Thousand Oaks, CA: Sage Publications, Inc.

Bennett, K. B., & Flach, J. M. (1992). Graphical displays: Implications for divided attention, focused attention, and problem solving. *Human Factors, 34*(5), 513–533.

Berner, E., Brooks, C., Miller, R., Masarie, F., & Jackson, J. (1989). Evaluation issues in the development of expert systems in medicine. *Evaluation and the Health Professions, 12*, 270–281.

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*. Hillsdale, NJ: Lawrence Erlbaum.

Bisantz, A. M., Roth, E., & Brickman, B. (2003). Integrating cognitive analysis in a large-scale system design process. *International Journal of Human-Computer Studies, 58*, 177–206.

Bittner, K. (2002). *Use case modeling*. Harlow, England: Addison-Wesley Publishing.

Bjork, R. A. (1999). Assessing our own competence: Heuristics and illusions. *Attention and Performance, 17*, 435–459.

Blank, H. (1998). Memory states and memory tasks: An integrative framework for eyewitness memory and suggestibility. *Memory, 6*(5), 481–529.

Blount, S., & Larrick, R. P. (2000). Framing the game: Examining frame choice in bargaining. *Organizational Behavior and Human Decision Processes, 81*(1), 43–71.

Bochman, A. (2004). A causal approach to nonmonotonic reasoning. *Artificial Intelligence, 160*(1–2), 105–143.

Bodker, K., Kensing, F., & Simonsen, J. (2004). *Participatory IT design: Designing for business and workplace realities*. Cambridge, MA: MIT Press.

Bowers, C., Salas, E., & Jentsch, F. (Eds.). (2006). *Creating high-tech teams: Practical guidance on work performance and technology*. Washington, DC: American Psychological Association.

Brachman, R., & Levesque, H. (2004). *Knowledge representation and reasoning*. San Mateo, CA: Morgan Kaufmann.

Bradfield, A., & Wells, G. (2005). Not the same old hindsight bias: Outcome information distorts a broad range of retrospective judgments. *Memory and Cognition, 33*(1), 120–130.

Brehm, S., Kassin, S. M., & Fein, S. (1999). *Social psychology* (4th ed.). Boston: Houghton Mifflin Company.

Burns, C., Bizantz, A., & Roth, E. (2004). Lessons from a comparison of work domain models: Representational choices and their implications. *Human Factors, 46*, 711–727.

Burns, C., & Hajdukiewicz, J. (2004). *Ecological interface design*. Boca Raton, FL: CRC Press.

Burns, C., & Vicente, K. (2001). Model-based approaches for analyzing cognitive work: A comparison of abstraction hierarchy, multi-level flow modeling, and decision ladder modeling. *International Journal of Cognitive Ergonomics, 5*, 357–366.

Byrne, M., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology, 15*(2), 135–155.

Caggiano, D., & Parasuraman, R. (2004). The role of memory representation in the vigilence decrement. *Psychonomic Bulletin and Review, 11*(5), 932–937.

Caldwell, B. (2005). Multi-team dynamics and distributed expertise in mission operations. *Aviation, Space, and Environmental Medicine, 76*(6), 145–153.

Card, S., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Carroll, J. M. (1995). *Scenario-based design: Envisioning work and technology in system development*. New York: John Wiley and Sons.

Carroll, J. M., Neale, D. C., Isenhour, P. L., & McCrickard, D. S. (2003). Notification and awareness: Synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies, 58*, 605–632.

Cartwright, D., & Zander, A. (Eds.). (1960). *Group dynamics: Research and theory* (2nd ed.). Evanston, IL: Row and Peterson.

Cary, M., & Carlson, R. (2001). Distributing working memory resources during problem solving. *Journal of Experimental Psychology: Learning, Memory and Cognition, 27*, 836–848.

Chandrasekaran, B. (1988). Generic tasks as building blocks for knowledge based systems: The diagnosis and routine design examples. *Knowledge Engineering Review, 3*(3), 183–210.

Charness, N., Feltovich, P., Hoffman, R., & Ericsson, E. (2006). *The Cambridge handbook of expertise and expert performance.* New York: Cambridge University Press.

Clancey, W. J. (1983). The epistemology of a rule-based expert system—a framework for explanation. *Artificial Intelligence, 20*, 215–251.

Clancey, W. J. (1985). Heuristic classification. *Artificial Intelligence, 27*, 289–350.

Clarke, K., Hardstone, G., Rouncefield, M., & Sommerville, I. (Eds.). (2006). *Trust in technology: A socio-technical perspective.* New York: Springer.

Cockburn, A. (2000). *Writing effective use cases*. Harlow, England: Addison-Wesley.

Crandall, B., Klein, G., & Hoffman, R. (2006). *Working minds: A practitioner's guide to cognitive task analysis*. Cambridge, MA: MIT Press.

Croson, R., & Sundali, J. (2005). The gambler's fallacy and the hot hand: Empirical data from casinos. *Journal of Risk and Uncertainty, 30*(3),195–209.

Dasgupta, P., Chakrabarti, P., & Desarkar, S. (2004). *Multiobjective heuristic search: An introduction to intelligent search methods for multicriteria optimization*. San Mateo, CA: Morgan Kauffmann.

Dechter, R. (2003). *Constraint processing*. San Mateo, CA: Morgan Kaufmann.

Diaper, D., & Stanton, N. (Eds.). (2004). *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum.

Eckstein, M. P. (2004). Active vision: The psychology of looking and seeing. *Perception, 33*(8), 1021–1023.

Elmore, W. K., Dunlap, R. D., & Campbell, R. H. (2001). Features of a distributed intelligent architecture for unmanned air vehicle operations. *Proceedings of 2001 Unmanned Systems International, AUVSI*, Baltimore, MD.

Elstein, A. S., Shulman, L. S., & Sprafka, S. A. (1978). *Medical problem solving: An analysis of clinical reasoning*. Cambridge, MA: Harvard University Press.

Endsley, M. (2003). *Designing for situation awareness*. London: Taylor & Francis.

Endsley, M., & Garland, D. (2000). *Situation awareness analysis and measurement*. Mahwah, NJ: Lawrence Erlbaum Associates.

Erkens, G., Andriessen, J., & Peters, N. (2003). Interaction and performance in computer-supported collaborative tasks. In H. Van Oostendorp (Ed.), *Cognition in a digital world* (pp. 225–251). Mahwah, NJ: Lawrence Erlbaum.

Flach, J., Hancock, P., Caird, J., & Vicente, K. (Eds.). (1995). *Global perspectives on the ecology of human-machine systems*. Hillsdale, NJ: Lawrence Erlbaum.

Fischer, G., Lemke, A. C., Mastaglio, T., & Morch, A. I. (1991). The role of critiquing in cooperative problem solving. *ACM Transactions on Information Systems, 9*(3), 123–151.

Forsyth, D. (1998). *Group dynamics*. Florence, KY: Wadsworth Publishing.

Fraser, J. M., Smith, P. J., & Smith, J. W. (1992). A catalog of errors. *International Journal of Man-Machine Systems, 37*, 265–307.

Freeman, F., Mikulka, P., & Scerbo, M. (2004). An evaluation of an adaptive automation system using a cognitive vigilance task. *Biological Psychology, 67*(3), 283–297.

Garb, H. (2005). Clinical judgment and decision making. *Annual Review of Clinical Psychology, 1*(1), 67–89.

Geddes, N. D. (1997a). Large scale models of cooperating and hostile intents. *Proceeding of the 1997 IEEE Conference on Engineering of Computer Based Systems*, Monterey, CA, 94–100.

Geddes, N. D. (1997b). Associate systems: A framework for human-machine cooperation. In Smith, M. J., Salvendy, G., & Koubek, R. J. (Eds.), *Advances in human factors/ergonomics: Vol. 21. Design of computing systems: Social and ergonomic considerations* (pp. 237–242). New York: Elsevier.

Geddes, N., & Lee, J. (1998). Intelligent control for automated vehicles: A decision aiding method for coordination of multiple uninhabited tactical aircraft. *Proceedings of 1998 Unmanned Systems International, AUVSI,* Huntsville, AL, 46–51.

Geddes, N. D., & Lizza, C. S. (1999). *Shared plans and situations as a basis for collaborative decision making in air operations* (SAE Paper 1999-01-5538). Paper presented at the SAE World Aeronautics Conference, Atlanta, GA.

Gentner, D., & Stevens, A. (1983). *Mental models*. Hillsdale, NJ: Lawrence Erlbaum.

Ghallib, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Mateo, CA: Morgan Kaufmann Publishers.

Gilovich, T., Griffin, D., & Kahneman, D. (Eds.). (2002). *Heuristics and biases: The psychology of intuitive judgment*. New York: Cambridge University Press.

Gilovich, T., Vallone, R., & Tversky, A. (1985). The hot hand in basketball: On the misperception of random sequences. *Cognitive Psychology, 17*, 295–314.

Gordon, A. (2004). The representation of planning strategies. *Artificial Intelligence, 153*(1/2), 287–305.

Gordon, S., & Gill, R. T. (1992). Knowledge acquisition with question probes and conceptual graph structures. In T. Lauer, E. Peacock, & A. Graesser (Eds.), *Questions and information systems* (pp. 29–46). Hillsdale, NJ: Lawrence Erlbaum.

Gordon, S. E., & Gill, R. T. (1997). Cognitive task analysis. In C. Zsambok & G. Klein (Eds.), *Naturalistic decision making* (pp. 131–140). Hillsdale, NJ: Lawrence Erlbaum.

Grabowski, M., & Sanborn, S. (2003). Human performance and embedded intelligent technology in safety-critical systems. *International Journal of Human-Computer Studies, 58*(6), 637–670.

Grier, R., Warm, J., & Dember, W. (2003). The vigilance decrement reflects limits in effortful attention, not mindlessness. *Human Factors, 45*(3), 349–359.

Griffith, T. L., Sawyer, J. E., & Neale, M. A. (2003). Virtualness and knowledge in teams: Managing the love triangle of organizations, individuals and information technology. *MIS Quarterly, 27*, 265–287.

Grondin, S. (2005). Overloading temporal memory. *Journal of Experimental Psychology: Human Perception and Performance, 31*(5), 869–879.

Guerlain, S., Smith, P. J., Obradovich, J. H., Rudmann, S., Strohm, P., Smith, J. W. et al. (1996). Dealing with brittleness in the design of expert systems for immunohematology. *Immunohematology, 12*, 101–107.

Guerlain, S., Smith, P. J., Obradovich, J. H., Rudmann, S., Strohm, P., Smith, J. W. et al. (1999). Interactive critiquing as a form of decision support: An empirical evaluation. *Human Factors, 41*, 72–89.

Hammer, J. M., & Singletary, B. A. (2004). Common shortcomings in software for autonomous systems. *Proceedings of the AIAA First Intelligent Systems Conference*, Chicago, IL, 19–24.

Handberg, R. (1995). Expert testimony on eyewitness identification: A new pair of glasses for the jury. *American Criminal Law Review, 32*(4), 1013–1064.

Handy, C. (1995). Trust and the virtual organization. *Harvard Business Review, 73*, 40–50.

Harris, S., & Owens, J. (1986). Some critical factors that limit the effectiveness of machine intelligence technology in military systems applications. *Journal of Computer-Based Instruction, 13*, 30–34.

Haselton, M. G., Nettle, D., & Andrews, P. W. (2005). The evolution of cognitive bias. In D. M. Buss (Ed.), *The handbook of evolutionary psychology* (pp. 724–746). Hoboken, NJ: John Wiley and Sons.

Hasling, D. W., Clancey, W. J., & Rennels, G. (1984). Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Systems, 20*, 3–19.

Hayes-Roth, B., & Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science, 3*(4), 275–310.

Helander, M., Landauer, T., & Prabhu, P. (Eds.). (1997). *Handbook of human-computer interaction*. Amsterdam: Elsevier.

Hertel, G., Geister, S., & Konradt, U. (2005). Managing virtual teams: A review of current empirical research. *Human Resource Management Review, 15*(1) 69–95.

Hester, R., & Garavan, H. (2005). Working memory and executive function: The influence of content and load on the control of attention. *Memory and Cognition, 33*(2), 221–233.

Hinds, P., & Kiesler, S. (Eds.). (2005). *Distributed work*. Cambridge, MA: MIT Press.

Hinds, P. J., & Mortensen, M. (2005). Understanding conflict in geographically distributed teams: The moderating effects of shared identity, shared context, and spontaneous communication. *Organization Science, 16*, 290–307.

Hoc, J.-M. (2000). From human-machine interaction to human-machine cooperation. *Ergonomics, 43*, 833–843.

Hollnagel, E. (1990). Responsibility issues in intelligent decision support systems. In D. Berry & A. Hart (Eds.), *Expert systems: Human issues*, Cambridge, MA: MIT Press.

Hollnagel, E. (2003). *Handbook of cognitive task design*. Mahwah, NJ: Lawrence Erlbaum.

Hollnagel, E., & Woods, D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering.* Boca Raton, FL: Taylor & Francis.

Horton, F., & Lewis, D. (Eds.). (1991). *Great information disasters*. London: Association for Information Management.

Hutchins, E. (1990). The technology of team navigation. In J. Galegher, R. Kraut, & C. Egido (Eds.), *Intellectual teamwork: Social and technical bases of collaborative work*. Hillsdale, NJ: Lawrence Erlbaum.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Jagacinski, R. J., & Flach, J. M. (2003). *Control theory for humans: Quantitative approaches to modeling performance*. Hillsdale, NJ: Lawrence Erlbaum.

Janis, I. (1982). *Groupthink* (2nd ed.). Boston: Houghton Mifflin.

Johnson, P., Duran, A., Hassebrock, F., Moller, J., Prietulla, M., Feltovich, P. et al. (1981). Expertise and error in diagnostic reasoning. *Cognitive Science, 5*, 235–283.

Johnston, W. A., & Dark, V. J. (1986). Selective attention. *Annual Review of Psychology, 37*, 43–75.

Jonassen, D., Tessmer, M., & Hannum, W. (1999). *Task analysis methods for instructional design*. Mahwah, NJ: Lawrence Erlbaum.

Jones, P., & Jacobs, J. (2000). Cooperative problem solving in human-machine systems: Theory, models and intelligent associate systems. *IEEE Transactions on Systems, Man and Cybernetics, 30*(4), 397–407.

Jones, E., & Nisbett, R. (1971). The actor and the observer: Divergent perceptions of the causes of behavior. In E. Jones, D. Kanouse, H. Kelley, R. Nisbett, S. Valins, & B. Weiner (Eds.), *Attributions: Perceiving the causes of behavior* (pp. 79–94). Morristown, NJ: General Learning Press.

Jones, P. M., & Mitchell, C. M. (1995). Human-computer cooperative problem solving: Theory, design, and evaluation of an intelligent associate system. *IEEE Transactions on Systems, Man, and Cybernetics, 25*, 1039–1053.

Jones, D. R., & Schkade, D. A. (1995). Choosing and translating between problem representations. *Organizational Behavior and Human Decision Processes, 61*(2), 214–223.

Josephson, J., & Josephson, S. (1994). *Abductive inference*. New York: Cambridge University Press.

Kahneman, D., & Tversky, A. (Eds.). (2000). *Choices, values and frames*. New York: Cambridge University Press.

Katz, N., Lazer, D., Arrow, H., & Contractor, N. (2004). Network theory and small groups. *Small Group Research, 35*(3), 307–332.

Kieras, D. (1991). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *Handbook of human-computer interaction* (2nd ed.; pp. •••). Amsterdam: Elsevier Science Publishers.

Kieras, D. (2004). GOMS models for task analysis. In D. Diaper & N. Stanton (Eds.), *The handbook of task analysis for human-computer interaction* (pp. 83–116). Mahwah, NJ: Lawrence Erlbaum.

Kieras, D., & Meyer, D. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391–438.

Kirwan, B., & Ainsworth, L. (1992). *A guide to task analysis*. London: Taylor & Francis.

Klayman, J., & Ha, Y. (1987). Confirmation, disconfirmation, and information in hypothesis testing. *Psychological Review, 94*, 211–228.

Klein, G. A. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G. Klein, J. Oransanu, R. Calderwood, & C. Zsambok, (Eds.), *Decision making in action: Models and method* (pp. 138–147). Norwood, NJ: Ablex.

Klein, G. (2000). Cognitive task analysis of teams. In J. M. Schraagen, S. Chipman, & V. Shalin (Eds.), *Cognitive task analysis* (417–430). Mahwah, NJ: Lawrence Erlbaum.

Klein, G., Caulderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man and Cybernetics, 19*, 462–472.

Kleinmuntz, D. N., & Schkade, D. A. (1993). Information displays and decision processes. *Psychological Science, 4*(4), 221–227.

Koehler, J. J. (1996). The base rate fallacy reconsidered: Descriptive, normative, and methodological challenges. *Behavioral and Brain Sciences, 9*(1), 1–53.

Kohler, W. (1992). *Gestalt psychology: An introduction to new concepts in modern psychology*. New York: Liveright Publishing Corporation.

Kolodner, J. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology, 17*, 248–294.

Kulak, D., & Guiney, E. (2003). *Use cases: Requirements in context* (2nd ed.). Harlow, England: Addison-Wesley Publishing.

Kushleyeva, Y., Salvucci, D., Lee, F., & Schunn, C. (2005). Deciding when to switch tasks in time-critical multitasking. *Cognitive Systems Research, 6*(1), 41–49.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1–64.

Larkin, J. H. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. •••). Hillsdale, NJ: Lawrence Erlbaum.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65–99.

Larson, A., & Hayes, C. (2005). An assessment of WEASEL: A decision support system to assist in military planning. *Proceedings of the 2005 Annual Meeting of the Human Factors and Ergonomics Society,* Orlando, 287–291.

Lee, J., & See, K. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors, 46*(1), 50–80.

Lehner, P. E., & Zirk, D. A. (1987). Cognitive factors in user/expert-system interaction. *Human Factors, 29*(1), 97–109.

Lesgold, A., Glaser, R., Rubinson, H., Klopfer, D., Feltovich, P., & Wang, Y. (1988). Expertise in a complex skill: Diagnosing X-ray pictures. In M. Chi, R. Glaser, & M. Farr (Eds.), *The nature of expertise* (pp. 311–342). Hillsdale, NJ: Lawrence Erlbaum.

Levi, D. (2001). *Group dynamics for teams*. London: SAGE Publications.

Lewis, C., & Wharton, C. (1997). Cognitive walkthroughs. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed.; pp. 717–731). Amsterdam: Elsevier.

Lizza, C. S., & Banks, S. B. (1991, June). Pilot's associate: A cooperative knowledge based system application. *IEEE Expert*, *8*, 24–32.

Loftus, E. (1975). Leading questions and the eyewitness report. *Cognitive Psychology, 7*, 560–572.

Logan, G. (2005). Attention, automaticity, and executive control. In A. F. Healy (Ed.), *Experimental cognitive psychology and its applications* (pp. 129–139). Washington, DC: American Psychological Association.

Lowe, R. K. (2003). Animation and learning: Selective processing of information in dynamic graphics. *Learning and Instruction, 13*(2), 157–176.

Luo, Y., Greenwood, P., & Parasuraman, R. (2001). Dynamics of the special scale of visual attention revealed by brain event-related potentials. *Cognitive Brain Research, 12*(3), 371–381.

Lurey, J. S., & Raisinghani, M. S. (2001). An empirical study of best practices in virtual teams. *Information Management, 38*, 523–544.

Mack, R. (1995). Scenarios as engines of design. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 361–386). New York: John Wiley.

Mackworth, N. (1961). Researches on the measurement of human performance. In H. W. Sinaiko (Ed.), *Selected papers on human factors in the design and use of control systems* (pp. 48–63). New York: Dover Publications. (Original work published 1950)

Madani, O., Condon, A., & Hanks, S. (2003). On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence, 147*(1/2), 5–34.

Majercik, S., & Littman, M. (2003). Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence, 147*(1/2), 119–62.

Marsh, S., & Dibben, M. (2003). The role of trust in information science and technology. *Annual Review of Information Science and Technology, 37*, 465–498.

McCauley, C. (1989). The nature of social influence in groupthink: Compliance and internalization. *Journal of Personality and Social Psychology, 22*, 250–260.

Meister, D., & Enderwick, T. (2002). *Human factors in system design, development and testing.* Mahwah, NJ: Lawrence Erlbaum.

Metzger, U., & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human Factors, 47*(1), 35–49.

Meyers, D. (2004). *Psychology* (7th ed.). New York: Worth Publishers.

Michalewicz, Z., & Fogel, D. (2004). *How to solve it: Modern heuristics*. New York: Springer.

Michie, D. (1986). *On machine intelligence* (2nd ed.). Chicester: Ellis Horwood Limited.

Mikulka, P., Scerbo, M., & Freeman, F. (2002). Effects of a biocybernetic system on vigilance performance. *Human Factors, 44*(4), 654–664.

Miller, C., Pelican, M., & Goldman, R. (2000). "Tasking" interfaces to keep the operator in control. *Proceedings of the 5th International Conference on Human Interaction with Complex Systems*, Urbana, IL. 116–122.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior.* New York: Henry Holt and Company.

Miller, P. (1986). *Expert critiquing systems: Practice-based medical consultation by computer.* New York: Springer-Verlag.

Mital, A., & Pennathur, A. (2004). Advanced technologies and humans in manufacturing workplaces: An interdependent relationship. *International Journal of Industrial Ergonomics, 33*(4), 295–313.

Mitchell, C. M., & Miller, R. A. (1986). A discrete control model of operator function: A methodology for information display design. *IEEE Transactions on Systems. Man and Cybernetics, 16*, 343–357.

Muir, B. (1987). Trust between humans and machines. *International Journal of Man-Machine Studies, 27*, 527–539.

Muir, B., & Moray, N. (1996). Trust in automation 2: Experimental studies of trust and human intervention in a process control simulation. *Ergonomics, 39*(3), 429–460.

Mumford, M., Schultz, R., & Van Doorn, J. (2001). Performance in planning: Processes, requirement, and errors. *Review of General Psychology, 5*(3), 213–240.

Mynatt, C., Doherty, M., & Tweney, R. (1978). Confirmation bias in a simulated research environment: An experimental study of scientific inference. *Quarterly Journal of Experimental Psychology, 30*, 85–95.

Nakatsu, R. T., & Benbasat, I. (2003). Improving the explanatory power of knowledge-based systems: An investigation of content and interface-based enhancements. *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans, 33*(3), 344–357.

Nareyek, A., Fourer, R., & Freuder, E. (2005). Constraints and AI planning. *IEEE Intelligent Systems, 20*(2), 62–72.

Nason, S., Laird, J., & Schunn, C. (2005). SOAR-RL: Integrating reinforcement learning with SOAR. *Cognitive Systems Research, 6*(1), 51–59.

Navon, D. (1978). The importance of being conservative. *British Journal of Mathematical and Statistical Psychology, 31*, 33–48.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall, Inc.

Nickerson, R. (1988). Counting, computing, and the representation of numbers. *Human Factors, 30*, 181–199.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88*(1), 1–15.

Norman, D. A. (2002). *The design of everyday things*. New York: Doubleday.

Oliver, D., & Roos, J. (2005). Decision making in high-velocity environments. *Organization Studies, 26*(6), 889–913.

Olson, G. M., & Olson, J. S. (1997). Research on computer supported cooperative work. In M. Helander, T. Landauer, P. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 1433–1456). Amsterdam: Elsevier.

Olson, G. M., & Olson, J. S. (2003). Groupware and computer-supported cooperative work. In A. Sears & J. Jacko (Eds.), *Handbook of human-computer interaction* (pp. 583–593). Mahwah, NJ: Lawrence Erlbaum.

Orasanu, J., & Salas, E. (1991). Team decision making in complex environments. In G. Klein, R. Calderwood, & C. Zsambok (Eds.), *Decision making in action: Models and methods* (pp. 327–345). New Jersey: Ablex.

Ormrod, J. (2003). *Human learning* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.

Parasuraman, R. (2000). Designing automation for human use: Empirical studies and quantitative models. *Ergonomics, 43*, 931–951.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse and abuse. *Human Factors, 39*, 230–253.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems Man and Cybernetics, 30*(3), 286–297.

Pashler, H., Johnston, J. C., & Ruthruff, E. (2001). Attention and performance. *Annual Review of Psychology, 52*, 629–651.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufman.

Pennington, N., & Hastie, R. (1992). Explaining the evidence: Tests of the story model for juror decision making. *Journal of Personality and Social Psychology, 62*, 189–206.

Petroski, H. (1994). *Design paradigms: Case histories of error and judgment in engineering*. New York: Cambridge University Press.

Plous, S. (1993). *The psychology of judgment and decision making*. New York: McGraw-Hill.

Poulton, E. C. (1989). *Bias in quantifying judgments*. Hillsdale, NJ: Lawrence Erlbaum.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design*. New York: John Wiley and Sons.

Preece, J., Rogers, Y., Sharp, H., & Benyon D. (1994). *Human-computer interaction*. Addison-Wesley.

Pritchett, A., Vandor, B., & Edwards, K. (2002). Testing and implementing cockpit alerting systems. *Reliability Engineering and System Safety, 75*(2), 193–206.

Psillos, S. (2002). Simply the best: A case for abduction. *Lecture Notes in Artificial Intelligence, 2408*, 605–625.

Radvansky, G. (2005). *Human memory*. Boston: Allyn and Bacon.

Rasmussen, J. (1983). Skills, rules and knowledge: Signals, signs, symbols and other distinctions in human performance models. *IEEE Transactions on Systems Man and Cybernetics, 13*(3), 257–266.

Rasmussen, J., Brehner, B., & Leplat, J. (Eds.). (1991). *Distributed decision making: Cognitive models for cooperative work*. New York: John Wiley and Sons.

Rasmussen, J., Pejtersen, A., & Goldstein, L. (1994). *Cognitive systems engineering*. New York: John Wiley & Sons.

Rayward-Smith, V., Osman, I., Reeves, C., & Smith, G. (Eds.). (1996). *Modern heuristic search methods*. New York: John Wiley and Sons.

Reason, J. (1991). *Human error*. New York: Cambridge Press.

Reason, J. (1997). *Managing the risks of organizational accidents*. Hampshire, UK: Ashgate.

Reising, D., & Sanderson, P. (2002). Work domain analysis and sensors I: Principles and simple example. *International Journal of Human-Computer Studies, 56*(6), 569–596.

Riegelsberger, J., Sasse, M., & McCarthy, J. (2005). The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies, 62*(3), 381–422.

Riesbeck, C. K., & Schank, R. C. (1989). *Inside case-based reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Roth, E. M., Bennett, K. B., & Woods, D. D. (1987). Human interaction with an 'intelligent' machine. *International Journal of Man-Machine Studies, 27*, 479–525.

Rouse, W.B. (1980). *Systems engineering models of human machine interaction*. New York: Elsevier.

Rouse, W. B., Geddes, N. D., & Hammer, J. M. (1990, March). Computer-aided fighter pilots. *IEEE Spectrum, 8,* 38–41.

Rubin, K. S., Jones, P. M., & Mitchell, C. M. (1988). OFMSpert: Interference of operator intentions in supervisory control using a blackboard structure. *IEEE Transactions on Systems Man and Cybernetics, 18*(4), 618–637.

Russell, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence, 5*(2), 115–135.

Salas, E., Bowers, C., & Edens, E. (Eds.). (2001). *Improving teamwork in organizations: Applications of resource management training*. Mahwah, NJ: Lawrence Erlbaum.

Salas, E., & Fiore, S. (Eds.). (2004). *Team cognition: Understanding the factors that drive process and performance*. Washington, DC: American Psychological Association.

Sarter, N., & Woods, D. (1993). *Cognitive engineering in aerospace applications: Pilot interaction with cockpit automation* (NASA Contractor Rep. 177617). Moffett Field, CA : NASA Ames Research Center.

Scaife, M., & Rogers, Y. (1996). External cognition: How do graphical representations work? *International Journal of Human-Computer Studies, 45*, 185–213.

Schaeken, W. (Ed.). (2005). *Mental models theory of reasoning: Refinements and extensions*. Mahwah, NJ: Lawrence Erlbaum.

Schragen, J. M., Chipman, S., & Shalin, V. (Eds.). (2000). *Cognitive task analysis*. Mahwah, NJ: Lawrence Erlbaum.

Schroeder, R., & Axelsson, A.-S. (Eds.). (2005). *Work and plan in shared virtual environments: Computer supported cooperative work.* New York: Springer.

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory design: Principles and practices*. Mahwah, NJ: Lawrence Erlbaum.

Scott, R., Roth, E. M., & Deutsch, S. E. (2005). Work-centered support systems: A human-centered approach to intelligent system design. *IEEE Intelligent Systems, 20*, 73–81.

Seamon, J. (2005). *Human memory: Contemporary readings*. Cambridge: Oxford University Press.

Sewell, D. R., & Geddes, N. D. (1990). A plan and goal based method for computer-human system design. *Proceedings of INTERACT 90,* New York, 283–288.

Shafer, G. (1996). *Probabilistic expert systems*. Philadelphia: Society for Industrial and Applied Mathematics.

Shafer, G., & Pearl, J. (Eds.). (1990). *Readings in uncertain reasoning*. San Mateo, CA: Morgan Kaufmann.

Shalin, V. L., Geddes, N. D., Bertram, D., Szczepkowski, M. A., & DuBois, D. (1997). Expertise in dynamic physical task domains. In P. Feltovich, K. Ford, & R. Hoffman (Eds.), *Expertise in context: Human and machine* (pp. 194–217). Cambridge, MA: MIT Press.

Shepherd, A. (2000). *Hierarchical task analysis.* London: Taylor & Francis.

Sheridan, T. B. (1997). Supervisory control. In G. Salvendy (Ed.), *Handbook of human factors* (2nd ed.; pp. 1295–1327). New York: John Wiley and Sons.

Sheridan, T. (2002). *Humans and automation: System design and research issues*. Chidester, England: Wiley.

Sheridan, T. B., & Ferrell, W. R. (1974). *Man-machine studies: Information, control, and decision models of human performance*. Cambridge, MA: MIT Press.

Sheridan, T. B., & Verplank, W. L. (1978). *Human and computer control of undersea teleoperators* (MIT Man Machine Systems Laboratory Tech. Rep 1978.). Cambridge, MA: MIT Man Machine Systems Laboratory.

Shortliffe, E. (1990). Clinical decision support systems. In E. Shortliffe & L. Perreault (Eds.), *Medical informatics: Computer applications in health care* (pp. 466–500). New York: Addison Wesley.

Silverman, B. G. (1992). Survey of expert critiquing systems: Practical and theoretical frontiers. *Communications of the ACM, 35*(4), 106–128.

Skirka, L., Mosier, K., & Burdick, M. (1999). Does automation bias decision making? *International Journal of Human-Computer Systems, 51*, 991–1006.

Smith, P. J., Beatty, R., Spencer, A., & Billings, C. (2003). Dealing with the challenges of distributed planning in a stochastic environment: Coordinated contingency planning. *Proceedings of the 2003 Annual Conference on Digital Avionics Systems*, Chicago, IL., 116–123.

Smith, P. J., Billings, C., Chapman, R., Obradovich, J. H., McCoy, E., & Orasanu, J. (2000). Alternative architectures for distributed cooperative problem-solving in the national airspace system. In M. Benedict (Ed.), *Proceedings of the 5th International Conference on Human Interaction with Complex Systems*, Urbana, IL, 87–93.

Smith, P. J., & Geddes, N. (2003). A cognitive systems engineering approach to the design of decision support systems. In A. Sears & J. Jacko (Eds.), *Handbook of human-computer interaction* (pp. 656–675). Mahwah, NJ: Lawrence Erlbaum.

Smith, P. J., Giffin, W., Rockwell, T., & Thomas, M. (1986). Modeling fault diagnosis as the activation and use of a frame system. *Human Factors, 28*(6), 703–716.

Smith, P. J., McCoy, E., & Layton, C. (1997). Brittleness in the design of cooperative problem-solving systems: The effects on user performance. *IEEE Transactions on Systems, Man, and Cybernetics, 27*(3), 360–371.

Smith, P. J., McCoy, E., Orasanu, J., Billings, C., Denning, R., Rodvold, M. et al. (1997). Control by permission: A case study of cooperative problem-solving in the interactions of airline dispatchers and ATCSCC. *Air Traffic Control Quarterly, 4*, 229–247.

Smith, P. J., McCoy, E., & Orasanu, J. (2000). Distributed cooperative problem-solving in the air traffic management system. In G. Klein and E. Salas (Eds.), *Naturalistic decision making* (pp. 369–384). Mahwah, NJ: Lawrence Erlbaum.

Smith, P. J., McCoy, E., & Orasanu, J. (2001). Distributed cooperative problem-solving in the air traffic management system. In G. Klein and E. Salas (Eds.), *Naturalistic Decision Making,* Mahwah NJ: Lawrence Erlbaum, 369–384.

Smith, P. J., Obradovich, J. H., Guerlain, S., Rudmann, S., Strohm, P., Smith, J., et al. (1998). Successful use of an expert system to teach diagnostic reasoning for antibody identification. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems,* 354–363.

Smith, P. J., & Rudmann, S. (2005). Clinical decision making and diagnosis: Implications for immunohematologic problem-solving. In S. Rudmann (Ed.) *Serologic problem-solving: A systematic approach for improved practice* (pp. 1–16). Bethesda, MD: AABB Press.

Smith, P. J., Stone, R. B., & Spencer, A. (2006). Design as a prediction task: Applying cognitive psychology to system development. In W. Marras & W. Karwowski (Eds.), *Handbook of industrial ergonomics* (2nd ed.). New York: Marcel Dekker, Inc.

St. Amant, R., Freed, A., Ritter, F. & Schunn, C. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research, 6*(1), 71–88.

Stenz, A., & Dias, M. B. (1999). *A free market architecture for coordinating multiple robots* (Tech. Rep. CMU-RI-TR-99-42). Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Strauch, B. (2004). *Investigating human error: Incidents, accidents, and complex systems*. Hampshire, UK: Ashgate Publishing.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication.* New York: Cambridge University Press.

Sycara, K., & Lewis, M. (2004). Integrating intelligent agents into human teams. In E. Salas & S. Fiore (Eds.), *Team cognition: Understanding the factors that drive process and performance* (pp. 203–231). Washington, DC: American Psychological Association.

Tetlock, P. (1998). Social psychology and world politics. In D. Gilbert, S. Fiske, & G. Lindzey (Eds.), *The handbook of social psychology: Vol. 2.* (4th ed.; pp. 868–912). New York: McGraw Hill.

Tetlock, P., Peterson, R., McQuire, M., Chang, S., & Feld, P. (1992). Assessing political group dynamics: A test of the groupthink model. *Journal of Personality and Social Psychology, 63*, 403–425.

Tianfield, H., & Wang, R.W. (2004). Critic systems: Towards human-computer collaborative problem solving. *Artificial Intelligence Review, 22*(4), 271–295.

Tobey, D. (2005). *Needs assessment basics*. Alexandria, VA: ASTD Press.

Turban, E., Aronson, J., & Liang, T.-P. (2004). *Decision support systems and intelligent systems* (7th ed.). Englewood Cliffs, NJ: Prentice Hall.

Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1990). *Envisioning information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1997). *Visual explanations*. Cheshire, CT: Graphics Press.

Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological Review, 79*, 281–299.

Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases*. New York: Cambridge University Press.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science, 185*, 1124–1131.

Vicente, K. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum.

Vicente, K. (2000). Work domain analysis and task analysis: A difference that matters. In J. M. Schraagen, S. Chipman, & V. Shalin (Eds.), *Cognitive task analysis* (pp. 101–118). Mahwah, NJ: Lawrence Erlbaum.

Vicente, K. J. (2002). Ecological interface design: Progress and challenges. *Human Factors, 44*(1), 62–78.

Vicente, K., & Rasmussen, J. (1990). The ecology of human-machine systems II: Mediating "direct perception" in complex work domains. *Ecological Psychology, 2*, 207–249.

Vicente, K., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics, 22*, 589–606.

Waltz, D. (1975). Understanding line drawings of scenes with shadows. In P. Winston (Ed.), *Psychology of computer vision* (pp. 29–45). Cambridge, MA: MIT Press.

Wason, P. (1960). On the failure to eliminate hypotheses in a conceptual task. *Quarterly Journal of Experimental Psychology, 12*, 129–140.

Watzman, S. (2003). Visual design principles for usable interfaces. In A. Sears & J. Jacko (Eds.), *Handbook of human-computer interaction* (pp. 263–285). Mahwah, NJ: Lawrence Erlbaum.

Wickens, C. D., Lee, J., Liu, Y., & Becker, S. G. (2004). *An introduction to human factors engineering* (2nd ed.). Upper Saddle River, NJ: Pearson/Prentice Hall.

Wiegmann, D., & Shappell, S. (2003). *A human error approach to aviation accident analysis: The human factors analysis and classification system*. Hampshire, UK: Ashgate.

Wiegmann, D., Zhang, H., Von Thaden, T., Sharma, G., & Gibbons, A. (2004). Safety culture: An integrative review. *International Journal of Aviation Psychology, 14*(2), 117–134.

Wiener, E. L., & Curry, R. E. (1980). Flight deck automation: Promises and problems. *Ergonomics, 23*(10). 995–1011.

Wiener, E. L., & Nagel, D. C. (1988). *Human factors in aviation*. New York: Academic Press.

Wilensky, R. (1983). *Planning and understanding: A computational approach to human reasoning*. Reading, MA: Addison-Wesley.

Williams, K. (2005). Computer-aided GOMS: A description and evaluation of a tool that integrates existing research for modeling human-computer interaction. *International Journal of Human-Computer Interaction, 1*(1), 39–58.

Witkin, B. R., & Altshuld, J. S. (1995). *Planning and conducting needs assessments*. Thousand Oaks, CA: Sage Publications.

Wogalter, M. S., & Mayhorn, C. B. (2005). Providing cognitive support with technology-based warning systems. *Ergonomics, 48*(5), 522–533.

Yantis, S. (1998). Control of visual attention. In H. Pashler (Ed.), *Attention* (pp. 223–256). Hillsdale, NJ: Lawrence Erlbaum.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21*(2), 179–217.

Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87–122.

Zlot, R., & Stenz, A. (2003, June 13–14). Multirobot control using task abstraction in a market framework. *Proceedings of the 4th International Conference on Field and Service Robotics*. 64–70.

Zuboff, S. (1988). *In the age of smart machines*. New York: Basic Books.

·30·

ONLINE COMMUNITIES

Panayiotis Zaphiris, Chee Siang Ang, and Andrew Laghos
City University, London

Introduction **604**
CMC and Online Communities **604**
Computer Mediated Communication 604
Online Communities 604
Examples of CMC and Online Communities 605
Wiki-based communities 606
Online virtual game communities 607
Analyzing Online Communities: Frameworks and Methodologies **608**
Query-Based Techniques and User Profiles: Interviews, Questionnaires, and Personas 609
Interviews 609
Questionnaires 609
Personas 609
Log Analysis 609
Content and Textual Analysis 610
Social Network Analysis (SNA) 610
Case Studies **612**
Computer Aided Language Learning Communities 612
Game Communities and Activity Theoretical Analysis 614
Types of game communities 614
Activity theory 614
Activity theoretical analysis for constructionist learning 616
Discussion and Conclusions **617**
References **618**

INTRODUCTION

The expansion of the Internet has resulted in an increase in the usefulness of Computer Mediated Communication (CMC) and the popularity of online communities. It is estimated that 25% of Internet users have participated in chat rooms or online discussions (Madden & Rainie, 2003).

This chapter begins with an introduction to the area of CMC and online communities by providing their definitions as well as their advantages and disadvantages. Different types of CMC are then introduced and two special types of online communities are described in more depth. More specifically, we pay special attention to the evolution of Wiki and game-based communities. These two relatively new areas of online sociability create new opportunities and challenges in the way people work, learn, and play online. Wiki-based communities facilitate new modes of social collaboration, where the creation of online content is no longer an individual action but rather it is transformed into a social, collaborative activity. Massively Multi-Player Online Role Playing Games (MMORPGs) have taken the social aspects of computer game playing to a new dimension, where players interact, socialize, and form networks of communities by having fun online.

Online communities are a source of valuable data that, when properly analyzed, can provide us with insights about the social experience people who are part of them have. For this reason, the analysis and evaluation of online communities requires a good understanding of all the available evaluation frameworks and methodologies that exist. We provide a description of the key methods later in this chapter. We then demonstrate the application of some of these methods to two characteristic case studies. Our first case study looks at how learning communities can be analyzed and how results from this analysis can be used for improving the pedagogical value of e-learning. The second case study investigates the use of activity theoretical frameworks in the analysis of computer-game-based communities. We then conclude this chapter with a brief summary and suggestions for new directions in the area of online communities.

CMC AND ONLINE COMMUNITIES

Computer Mediated Communication

It is by now no secret how vital the Internet was, is, and will continue to be in our lives. One of the most important characteristics of this medium is the opportunities it offers for human-human communication through computer networks. As Metcalfe (1992) pointed out, communication is the Internet's most important asset. E-mail is just one of the many modes of communication that can occur by using computers. Jones (1995) pointed out that through communication services, like the Internet, Usenet, and bulletin boards, online communication has for many people supplanted the postal service, telephone, and even the fax machine. All these applications in which the computer is used to mediate communication are called Computer-Mediated Communication (CMC).

December (1997) defined CMC as "the process by which people create, exchange, and perceive information using networked telecommunications systems (or non-networked computers) that facilitate encoding, transmitting, and decoding messages." He emphasized that studies of CMC view this process from different interdisciplinary theoretical perspectives (social, cognitive/psychological, linguistic, cultural, technical, and political) and often draw from fields such diverse as human communication, rhetoric and composition, media studies, human-computer interaction (HCI), journalism, telecommunications, computer science, technical communication, and information studies.

Online Communities

Online communities emerge by using CMC applications. The term *online community* is multidisciplinary in nature, means different things to different people, and is difficult to define (Preece, 2000). To gain a general understanding of what online communities are, Rheingold wrote that "[online] communities are social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace" (Rheingold, 1993, p. 5).

Online communities are also often referred to as cyber societies, cyber communities, web groups, virtual communities, web communities, virtual social networks, and e-communities, among several others.

Cyberspace is the new frontier in social relationships, and people are using the Internet to make friends, colleagues, lovers, as well as enemies (Suler, 2004). As Korzeny pointed out, even as early as 1978, online communities were formed around interests and not physical proximity (Korzeny, 1978). In general, what brings people together in an online community are common interests such as hobbies, ethnicity, education, and beliefs. As Wallace (1999) pointed out, meeting in online communities eliminates prejudging based on someone's appearance, and thus people with similar attitudes and ideas are attracted to each other.

It is estimated that as of September 2002 there are over 600 million people online (NICA Internet Surveys, 2004). The emergence of the so-called "global village" was predicted years ago (McLuhan, 1964) because of television and satellite technologies. However, it is argued by Fortner (1993) that "global metropolis" is a more representative term (Choi & Danowski, 2002). If one takes into account that the estimated world population of 2002 was 6.2 billion (U.S. Census Bureau, 2004), then the online population is nearly 10% of the world population—a significant percentage which must be taken into account when analyzing online communities. In most online communities, time, distance, and availability are no longer disseminating factors. Given that the same individual may be part of several different or numerous online communities, it is obvious why online communities keep increasing in numbers, size, and popularity.

Preece, Rogers, and Sharp (2002) stated that an online community consists of people, a shared purpose, policies, and computer systems. They identified the following member roles:

- Moderators and mediators who guide discussions/serve as arbiters;

- Professional commentators who give opinions/guide discussions;
- Provocateurs who provoke;
- General Participants who contribute to discussions;
- Lurkers who silently observe.

CMC has its benefits as well as its limitations. For instance, CMC discussions are often potentially richer than face-to-face discussions. However, users with poor writing skills may be at a disadvantage when using text-based CMC (ScotCIT, 2003). Table 30.1 summarized the advantages and disadvantages of CMC (ScotCIT, 2003).

Examples of CMC and Online Communities

Examples of CMC include asynchronous communication like e-mail and bulletin boards; synchronous communication like chatting; and information manipulation, retrieval, and storage through computers and electronic databases (Ferris, 1997). Table 30.2 lists the main types of CMC, their mode (synchronous or asynchronous), and the type of media they support (text, graphics, audio, video).

When it comes to website designers, choosing which CMC to employ (for instance, forum or chat room) is not a matter of luck or randomness. Selecting the right CMC tool depends on a lot of factors. For example, in the case of e-learning, the choice of the appropriate mode of CMC will be made by asking and answering questions such as (Bates, 1995; CAP, 2004; Heeren, 1996; Reiser & Gagné, 1983):

- Are the users spread across time zones? Can all participants meet at the same time?
- Do the users have access to the necessary equipment?
- What is the role of CMC in the course?
- Are the users good readers/writers?
- Are the activities time-independent?
- How much control is allowed to the students?

Audio conferencing is a real-time communication mechanism, since the communication happens synchronously. Depending on the application, text chat and graphics may also be supported. Video conferencing, like audio conferencing, offers a useful mode of communication but has the added benefit of allowing participants to see, not just hear, one another.

IRC and chats also support synchronous communication, since they enable the users to carry out conversations using text messaging. MUDs build on chats by providing avatars and graphical environments in which the users can engage in interactive fantasy games (Preece, 2000).

TABLE 30.1. Advantages and Disadvantages of CMC

Advantages
• Time and place independence • No need to travel • Time lapse between messages allows for reflection • Participants have added time to read and compose answers • Questions can be asked without waiting for a "turn" • It allows all participants to have a voice without the need to fight for "airtime," as in a face-to-face situation • The lack of visual cues provides participants with a more equal footing • Many to many interaction may enhance the communication • Answers to questions can be seen—and argued—by all • Discussion is potentially richer than in a face-to-face situation • Messages are archived centrally providing a database of interactions which can be revisited
Disadvantages
• Communication mainly takes place via written messages, so participants with poor writing skills may be at a disadvantage • Paralinguistic cues (facial expression, intonation, gesture, body orientation) as to a speakers' intention are not available, except through combinations of keystrokes (emoticons) or the use of typeface emphasis (italics, bold, capital letters) • Time gaps within exchanges may affect the pace and rhythm of communications leading to a possible loss in textual coherence • The medium is socially opaque; participants may not know which or how many people they may be addressing • The normal repair strategies of face-to-face communication are not available and misunderstandings may be harder to overcome • Context and reference of messages may be unclear and misunderstandings may occur

TABLE 30.2. CMC Systems, Their Mode, and the Types of Media that They Support

		Supports			
Type of CMC	Communication Mode	Text	Graphics	Audio	Video
Audio conferencing	Synchronous	Some applications	No	Yes	No
Video conferencing	Synchronous	Yes	Yes	Yes	Yes
IRC	Synchronous	Yes	as attachments	as attachments	as attachments
MUD	Synchronous	Yes	No	No	No
WWW	Sync & Async	Yes	Yes	Yes	Yes
E-mail	Asynchronous	Yes	as attachments	as attachments	as attachments
Newsgroups/BBS	Asynchronous	Yes	No	No	No
Discussion Boards	Asynchronous	Yes	as attachments	as attachments	as attachments
Voice mail	Asynchronous	Some applications	No	Yes	No
Wiki	Asynchronous	Yes	Yes	Yes	Possible
Online Virtual Game Environments	Synchronous	Yes	Yes	Yes	Yes

Websites are usually asynchronous, providing community information and links to other sites, but sometimes also have synchronous software, like chats, embedded in them (Preece, 2000).

E-mail is an asynchronous mode of communication, usually in the form of text. However, the ability to add attachments to e-mail messages makes it possible for audio, video, and graphic files to be used also. Voice mail is an expansion of e-mail whereby users may record themselves speaking out a message and then send that voice file to their contact, instead of typing it. Newsgroups, like e-mail, provide an asynchronous mode of communication, but unlike e-mail where the messages come to the users, it is a "pull" technology meaning the users must go to the UseNet groups themselves (Preece, 2000). Finally, discussion boards, also referred to as "forums" or "bulletin boards" provide an asynchronous mode of communication where the users post messages for others to see and respond on their own time.

Preece (2000) described in detail some of these types of CMC and their different characteristics. In this chapter, we focus on Wiki and Online Virtual Game Environments which in our view provide a set of new, novel modes of communication and online-community-building systems.

Wiki-based communities. The "Wiki", named for the Hawaiian word "quick," is a new technology that embodies the notion of new media use where everyone is the author. It is a freely expandable collection of hypertexts which can be easily edited by any user with knowledge of a simple markup language. It does not require any specific tools; all you need is a form-capable web-browser client. This simplistic method gives the freedom to everyone reading the page to amend or correct what he or she is reading. This aims to encourage people to contribute to the expansion of the page (Halvorsen, 2005). Every user can be an author by simply clicking the edit button, changing the current text, and submitting the new version, which is then converted automatically into HTML.

Wikis are a system that explicitly supports collaboration and community building as it decentralizes the effort of creating a website from the hands of the few and distributes it to a huge community of Internet users. In a Wiki environment, users are not only editing, they are also encouraged to create their own content and their own pages. A link to existing pages can be made easily, and a new page can be created by making a new link. Thus, apart from contents, the users also codesign the structure of a Wiki site.

The goal of Wiki sites is to become a shared repository of knowledge, with the knowledge base growing over time. Unlike chat rooms, Wiki content is expected to have some degree of seriousness and permanence. In a Wiki, the users create the content in collaboration and over time. Like weblogs, Wikis have been around for some time and are popular among the technology community. However, weblogs can be highly personal while Wikis are intensely collaborative. Whereas weblogs tend to use a modified WYSIWYG editing environment, Wikis use a simple set of formatting commands.

Recently there has been an increasing interest in using Wikis for learning (Wang et al., 2005; Jones, 2003). Although any knowledge-building application that demands the absolute and immutable integrity of the content is not really suitable for a Wiki, it is useful in situations where communities of people are developing shared ideas, values, or resources. A biology teacher for example could start a Wiki site by posting some material creating a tentative structure of the subject, and uploading some media files. When the students visit, they can expand the contents by modifying or posting more material, or making links to new pages, thus enriching the learning resources. Through shared construction cycles the students feel closer to the learning system as they contribute to its development instead of being passively presented with the information.

A project has already been undertaken to rebuild a web-based learning site for spectroscopy using Wiki technologies with the goal of making the resource more relevant and content-rich, to attract authors from different backgrounds to provide contents in multiple languages to support the international users (Mader, 2004). Perhaps the most famous educational Wiki in existence is Wikipedia (Wikipedia, 2005) a free-content encyclopedia in many languages that anyone can edit. It is the result of a vast collaboration and currently contains over a million entries in multiple languages. The Wikipedia project shows that the model might work, and that groups of people can collaboratively create shared knowledge artifacts.

However, a Wiki site could be highly unstructured because no editorial function examines the contributions or guarantees quality or accuracy of its content. It is the responsibility of the users to ensure correctness and it is their collective responsibility to take care of all aspects of policy, such as appropriate behavior in the community as well as rules (Halvorsen, 2005).

Since anyone can literally contribute anything, vandalism sometimes occurs within Wikis. Therefore, version control is crucial. Each version is saved, so that it is easy to reverse to a former version if necessary (Emigh & Herring, 2005). Studies have confirmed that vandals are often stopped and the vandalized pages are reversed quickly (Viégas, Wattenberg, & Dave, 2004; Aronsson, 2002).

Wiki-based collaboration could be a very practical platform for asynchronous distributed brainstorming; since Wikis allow ideas to be captured quickly, they facilitate elaboration on existing ideas and document the evolution of revisions made to ideas.

With the increasing popularity of Wikis, it is not surprising that recently it has been used to assemble online communities of various interests. The central access for all users has made the Wiki appropriate for project work, document production, the joint development of project concepts, and discussion forums of all types.

The Wiki technology appears to be a suitable approach for supporting cooperative community, especially for knowledge generation. One of the reasons that make it a successful tool for community building is its "anyone-can-contribute" policy, where producing new pages is quick and easy, as is linking them to existing pages.

Moreover, it also provides a simple means of communication through the discussion section associated with each Wiki page as well as the history entry which keeps track of the changes made by every user. In such a way, the users are constructing and discussing the page at the same time within the same space. This results in the rapid appearance of a chaotic network of Wiki

pages and the knowledge grows organically. Facilitated by the Wiki technology, users can contribute and integrate knowledge into and obtain knowledge from the existing database created by others.

The particular advantage of the Wiki approach compared to other CMC tools on knowledge generation and exchange systems is the focus on both the process as well as the result of communication (Kittowski, F. F. & Köhler, A., 2002).

Discussion boards focus largely on the process of the communication where opinions are exchanged, conflicts are resolved, and agreements are achieved regarding the documents/projects which are being worked on. The document/project itself is not being worked/expanded explicitly through the use of discussion boards; instead, only after the consensus is reached will the project/document be worked by extracting ideas from the postings of the discussion.

Content and document management systems are used to organize and facilitate collaborative creation of documents and other content, usually applications for maintaining documents, including version controls, right of access, document protections, and so on. In terms of collaboration, this allows the process of expansion for each document version by several people. Communication processes and discussions that lead to the final product are limited, for example, to annotations.

Wikis, in contrast, permit users to discuss and work on the document simultaneously. Moreover, cooperative production of content becomes very efficient through the realignment of the distinction between author and reader. Thus, the Wikis' features largely fulfill the requirements for a tool to support a cooperative, knowledge-building community.

In fact, many Wikis are collaborative communities. As mentioned previously, Wikis allow anyone to click the "edit button" and change the website. Surprisingly, many Wikis are able to do this successfully without major issues in terms of vandalism. One of the important vandal control features is that the Wiki stores the history of each page. Thus, for each vandal, there is probably a group of people who actually needs the information that was removed through vandalism, and who will reverse the page to its former version. In addition, some Wikis are in fact not completely open as they restrict access, while some even have a democratic error/vandalism reporting system.

Online virtual game communities. With the advent of ubiquitous broadband Internet connection and the increasing graphical processing power of personal computers, a new paradigm of gaming has emerged. Massively Multi-Player Online Role Playing Games (MMORPGs) have changed the game industry dramatically. MMORPGs provide a fictional setting where a large group of users voluntarily immerse themselves in a graphical virtual environment and interact with each other by forming a community of users.

Although the concept of multiplayer gaming is not new, the game world of most local network multiplayer games, as opposed to MMORPG, are simplistic and can accommodate only around 16 concurrent players in a limited space.

A MMORPG enables thousands of players to play in an evolving virtual world simultaneously over the Internet. The game world is usually modeled with highly detailed 3D graphics, allowing individuals to interact not only with the gaming environment, but also with other players. Usually this involves the players representing themselves with avatars, the visual representation of the player's identity in the virtual world.

The MMORPG environment is a new paradigm in computer gaming in which players are part of a persistent world which exists independent of the users (Yee, 2005). Unlike other games where the virtual world ceases to exist when players switch off the game, in an MMORPG, the world exists before the user logs on, and continues to exist when the user logs off. More importantly, events and interactions occur in the world even when the user is not logged on as many other players are constantly interacting, thus transforming the world. To accommodate the large number of users, the worlds in MMORPGs are vast and varied in term of "geographical locations," characters, monsters, items, and so forth. More often than not, new "locations" or items are added by the game developers from time to time according to the demand of the players.

On one hand, a MMORPG like a Role Playing Game (RPG); it involves killing monsters, collecting items, developing characters, and so on. However, it contains an extra aspect which is the internal sociability within the game. Unlike single player games which rely on other external modes of communication (such as mailing lists, discussion forums outside the game) to form the gaming culture, the culture is formed within the MMORPG environment itself.

In such a way, these MMORPG virtual worlds represent the persistent social and material world, which is structured around narrative themes (usually fantasy), where players are engaged in various activities: slaying monsters, attacking castles, scavenging goods, trading merchandise, and so on. On one hand, the game's virtual world represents the escapist fantasy; on another, it supports social realism (Kolbert, 2001).

That means games are no longer meant to be a solitary activity played by a single individual. Instead, the player is expected to join a virtual community that is parallel with the physical world, in which societal, cultural, and economical systems arise. It has been gradually becoming a world that allows players to immerse into experiences which closely match those of the real world: virtual relationships are sought, virtual marriages are held, virtual shops are set up, and so forth.

The MMORPG genre now boasts hundreds of thousands of users and accounts for millions of dollars in revenue each year. The number of people who play the games (and the time they invest in terms of activities within and around the game) is astounding. The MMORPG, Lineage (NCsoft, 2005), for example, had more than 2.5 million current subscribers in 2002 (Vaknin, 2002) and, within a year, Ultima Online (Electronic Art, 2005) attracted more than 160 million person-hours (Kolbert, 2001).

Such games are ripe for cultural analysis of the social practices around them. Although fundamentally MMORPGs are video games with virtual spaces with which the players interact, they should be regarded not just as a piece of game software; they are a community, a society, and, if you wish, a culture. These games are becoming the most interesting interactive computer-mediated communication and networked activity environment (Taylor, 2002). Thus, understanding the pattern of participation in these game communities is crucial, as such virtual

communities function as a major mechanism of socialization of the players to the norms of the game culture that emerges as Squire and Steinkeuhler (in press) noted:

> Playing one's character(s) and living in [these virtual worlds] becomes an important part of daily life. Since much of the excitement of the game depends on having personal relationships and being part of [the] community's developing politics and projects, it is hard to participate just a little (Squire, Steinkeuhler, in press)

Recently game designers have tried to stretch the boundaries further by structuring in-game activities to maximize interaction. One of the examples of sociability by design in MMORPG is Star Wars Galaxies (Sony, 2005) which is organized so that players are steered toward certain locations in the game world where social playing is expected to take place (Ducheneaut, Moore, & Nickell, 2004).

Such communities formed around the game can be broadly divided into two categories: in-game and out-of-game communities. Most MMORPGs are created to encourage long-term relationships among the players through the features that support the formation of in-game communities. One of the most evident examples is the concept of guilds. Guilds are a fundamental component of MMORPG culture for people who are natural organizers to run a virtual association which has formalized membership and rank assignments and encourage participation. Sometimes, a player might join a guild and get involved in a guild war in order to fight for the castle. Each guild usually has a leader and several guilds could team up in a war. This involves a complicated leader-subordinate and leader-leader relationship.

In addition, to encourage social interaction, MMORPGs are specially designed in such a way that some game goals are almost impossible to be achieved without forming communities. For example, one player alone could spend a long period collecting all of the items needed to assemble a device. But a guild could ask its members to fan out in small groups and collect all of the necessary components in one day. Complex devices beyond the reach of any individual player could be quickly constructed by the guild. The guild could also accept donations from members and then distribute those contributions to others according to their needs, benefiting everyone through this collaboration (Kelly, 2004).

Apart from relatively long-term relationship such as guild communities, MMORPGs also provide many opportunities for short-term relationship experiences. For example, a player could team up with another player to kill monsters in order to develop the abilities of their avatars ("level up") or some more expert players could help newer players to get through the game.

When trying to win the game, players often need to get information from other resources: guidebooks, discussion forums, other players, and the like. Therefore game playing is generally more concerned with player-player interaction than with player-game interaction. What is at first confined to the game alone soon spills over into the virtual world beyond it (e.g., websites, chatrooms, e-mail) and even life off-screen (e.g., telephone calls, face-to-face meetings).

Apart from these external communities around the game which are mediated through e-mails or online forums (which also exist in many other games), an interesting phenomenon fuses the internal and external game communities. The participation in an external community starts to break the magic circle of the game—that game space is no longer separate from real life—as the out-of-game community trades in-game items for real money.

For example, Norrath, the world of EverQuest, was estimated to have the 77th largest economy in the real world based on buying and selling in online auction houses.

> About 12,000 people call it their permanent home, although some 60,000 are present there at any given time. The nominal hourly wage is about USD3.42 per hour, and the labors of the people produce a GNP per capita somewhere between that of Russia and Bulgaria. A unit of Norrath's currency is traded on exchange markets at USD0.0107, higher than the Yen and the Lira. (Castronova, 2001)

Having illustrated the social phenomenon around such playful virtual community, it is believed that it is fruitful to research such communities as we might be able to derive some useful implications on how successful Computer Supported Collaborative Work (CSCW) and Computer Supported Collaborative Learning (CSCL) environments can be designed. For this reason, in the next section we will describe some of the methodologies that can be used in such studies, and later we will present the application of some of these methods to two case studies.

ANALYZING ONLINE COMMUNITIES: FRAMEWORKS AND METHODOLOGIES

There are various aspects and attributes of CMC that can be studied to help us better understand online communities—for instance, the analysis of the frequency of exchanged messages and the formation of social networks, or the analysis of the content of the exchanged messages and the formation of virtual communities. To achieve such an analysis a number of theoretical frameworks have been developed and proposed. For example, Henri (1992) provided an analytical model for cognitive skills that can be used to analyze the process of learning within messages exchanged between students of various online e-learning communities. Mason's work (1991) provided descriptive methodologies using both quantitative and qualitative analysis. Furthermore, five phases of interaction analysis are identified in Gunawardena, Lowe, and Anderson's model (1997):

1. Sharing/Comparing of Information
2. The Discovery and Exploration of Dissonance or Inconsistency among Ideas, Concepts, or Statements
3. Negotiation of Meaning/Co-Construction of Knowledge
4. Testing and Modification of Proposed Synthesis or Co-Construction
5. Agreement Statement(s)/Applications of Newly Constructed Meaning

In this section we provide a description of some of the most commonly used online community evaluation techniques as well as their weaknesses and strengths.

Query-Based Techniques and User Profiles: Interviews, Questionnaires, and Personas

Interviews. An interview can be defined as a type of conversation that is initiated by the interviewer in order to obtain research relevant information. The interview reports have to be carefully targeted and analyzed to make an impact (Usability Net, 2003). Interviews are usually carried out on a one-to-one basis in which the interviewer collects information from the interviewee. Interviews can take place by telephone or face to face (Burge & Roberts, 1993). They can also take place via nonreal-time methods like fax and e-mail, although in these cases they function like questionnaires. Interviews are useful for obtaining information that is difficult to elicit through approaches such as background knowledge and general principles. There are three types of interviews:

1. Structured interviews, which consist of predetermined questions asked in fixed order like a questionnaire;
2. Semi-structured interviews, in which questions are determined in advance but may be reordered, reworded, omitted, and elaborated;
3. Unstructured interviews, which are not based on predetermined questions but instead upon a general area of interest, thus allowing the conversation to develop freely.

Interviews can be used to gain insights about general characteristics of the participants of an online community and their motivations for participating in the community under investigation. The data collected comes straight from the participants of the online communities, whereby they are able to provide feedback based on their own personal experiences, activities, thoughts, and suggestions.

Advantages of Interviews (Usability Net, 2003) include (a) what is talked about can address directly the informant's individual concerns; (b) mistakes and misunderstandings can be quickly identified and cleared up; (c) they are more flexible than questionnaires; and (d) they can cover low-probability events.

Disadvantages of Interviews include (a) the danger of analyst bias towards his or her own knowledge and beliefs; (b) the accuracy and honesty of responses; and (c) they often must be used with other data-collection techniques to improve quality of data collected.

Questionnaires. A questionnaire is a self-reporting query-based technique. Questionnaires are typically produced on printed paper, but due to recent technologies and in particular the Internet, many researchers engage in the use of online questionnaires, thus saving time and money and eliminating the problem of a participant's geographical distance. Three types of questions can be used with questionnaires: (a) open questions, where the participants are free to respond however they like; (b) closed questions, which provide the participants with several choices for the answer; and (c) scales where the respondents must answer on a predetermined scale.

For online communities, questionnaire can be used to elicit facts about the participants, their behaviors, and their beliefs/attitudes. Like interviews, questionnaires are an important technique for collecting user opinions and experiences they have had using CMC and their overall existence in online communities.

The main advantages of questionnaires are that they are faster to carry out than observational techniques, and they can cover low-probability events. Disadvantages include (a) information is an idealized version of what should, rather than what does, happen; (b) responses may lack accuracy or honesty; (c) there is danger of researcher bias towards subset of knowledge he or she possesses; and (d) they must be used in conjunction with other techniques for validity.

Personas. Findings from interviews and questionnaires can be further used as a basis for developing user profiles using personas. A persona is a precise description of the user of a system, and of what he or she wishes to accomplish (Cooper, 1999). The specific purpose of a persona is to serve as a tool for software and product design. Although personas are not real people, they represent them throughout the design stage (Blomkvist, 2002) and are best based on real data collected through query-based techniques.

Personas are rich in details, including name, social history, and goals, and are synthesized from findings using query-based techniques with real people (Cooper, 1999). The technique takes into account user characteristics and creates a concrete profile of the typical user (Cooper, 1999).

For online communities, personas can be used to better understand the participants of the community and their backgrounds. Personas can also be used as a supplement to social network analysis (described later in this chapter) to get a greater overview of the characteristics of key participants of a community. Using personas, web developers gain a more complete picture of their prospective and/or current users and are able to design the interfaces and functionality of their systems to be more personalized and suited for the communication of the members of their online communities.

Advantages of personas include (a) they can be used to create user scenarios; (b) they can be anonymous, protecting use privacy; and (c) they represent the user stereotypes and characteristics.

Disadvantages of personas include that if not enough personas are used, users are forced to fall into a certain persona type which might now accurately represent them; and that they are time-consuming.

Log Analysis

A log, also referred to as weblog, server log, or log-file, is in the form of a text file and is used to track the users' interactions with the computer system they are using. The types of interaction recorded include key presses, device movements, and other information about the user activities. The data is collected and analyzed using specialized software tools and the range of data collected depends on the log settings. Logs are also time-stamped and can be used to calculate how long a user spends on a particular task or how long a user lingers in a certain part of the website (Preece, Rogers, & Sharp, 2002). In addition, an analysis of the server logs can help us find out when people visited the site, the areas they navigated, the length of their visits, the frequency of their visits, their navigation patterns, from where they are connected, and details about the computers they are using.

Log analysis is a useful and easy-to-use tool when analyzing online communities. For example, someone can use log analysis to answer more accurately questions like student attendance of

an online learning community. Furthermore, logs can identify the web pages users spend more time viewing, and the paths that they used. This helps identify the navigation problems of the website, but also gives a visualization of the users' activities in the virtual communities. For instance, in the case of e-learning communities, the log files will show which students are active in the CMC postings even if they are not active participants (few postings themselves), but just observing the conversations. Preece and Maloney-Krichmar (2003) noted that data logging does not interrupt the community, while at the same time can be used to examine mass interaction.

Advantages of logs (Preece et al., 2002) are that they (a) help evaluators analyze users' behavior; (b) help evaluators understand how users worked on specific tasks; (c) are unobtrusive; and (d) can automatically log large volumes of data.

Disadvantages of logs (Preece et al., 2002) are that (a) powerful tools are needed to explore and analyze the data quantitatively and qualitatively; and (b) there are user privacy issues.

Content and Textual Analysis

Content analysis is an approach to understanding the processes that participants engage in as they exchange messages (McLoughlin, 1996). There have been several frameworks created for studying the content of messages exchanged in online communities. Examples include work from Archer, Garrison, Anderson, and Rourke (2001); Gunawardena, Lowe, and Anderson's (1997) model for examining the social construction of knowledge in computer conferencing; Henri's (1992) content-analysis model and Fahy, Crawford, and Ally's (2001) Transcript Analysis Tool (TAT) which is described in more detail below.

The TAT focuses on the content and interaction patterns at the component level of the transcript (Fahy et al., 2001). After a lengthy experience with other transcript tools and reviews of previous studies, Fahy et al. (2001) chose to adapt Zhu's (1996) analytical model for the TAT. Zhu's model (1996) examined the forms of electronic interaction and discourse, the forms of participation and the direction of participant interaction in computer conferences. The TAT also contains echoes of Vygotskian theory, primarily those dealing with collaborative sense making, social negotiation, and proximal development (Cook & Ralston, 2003). The TAT developers have come up with the following strategic decisions (Fahy, 2003): (a) the sentence is the unit of analysis; (b) the TAT is the method of analysis; and (c) interaction is the criterion for judging conference success and topical progression (types and patterns).

The TAT was designed to permit transcript content to be coded reliably and efficiently (Fahy et al., 2001), while the advantages of TAT are (Fahy, 2003; Cook & Ralston, 2003; Fahy et al., 2001)

It reveals interaction patterns that are useful in assessing different communication styles and online behavioral preferences among participants;

It recognizes the complexity of e-conferences and measures the intensity of interaction;

It enables the processes occurring within the conferences to be noted and recorded;

It probes beyond superficial systems data, which mask the actual patterns of discussion;

It relates usefully to other work in the area;

It discriminates among the types of sentences within the transcript;

It reflects the importance of both social and task-related content and outcomes in transcript-analysis research.

The unit of analysis of the TAT is the sentence. In the case of highly elaborated sentences, the units of analysis can be independent clauses which, punctuated differently, could be sentences (Fahy, 2003). Fahy et al. (2001) have concluded that the selection of message-level units of analysis might partially explain problematic results that numerous researchers have had with previous transcript-analysis work. They also believe that the finer granularity of sentence-level analysis results in several advantages (Fahy, 2003):

Reliability;

Ability to detect and describe the nature of the widely varying social interaction, and differences in networking pattern, in the interactive behavior of an online community, including measures of social network density and intensity;

Confirmation of gender associations in epistolary/expository interaction patterns, and in the use of linguistic qualifiers and intensifiers. Table 30.3 shows the TAT categories (Fahy et al., 2001; Fahy, 2003).

Social Network Analysis (SNA)

Social Network Analysis (SNA) is the mapping and measuring of relationships and flows between people, groups, organizations, computers or other information/knowledge processing entities. The nodes in the network are the people and groups while the links show relationships or flows between the nodes. SNA provides both a visual and a mathematical analysis of human relationships. (Krebs, 2004, p. 1)

Preece (2000) added that SNA provides a philosophy and a set of techniques for understanding how people and groups relate to each other, and has been used extensively by sociologists (Wellman, 1982; Wellman, 1992), communication researchers (Rice, 1994; Rice, Grant, Schmitz, & Torobin, 1990) and others. Analysts use SNA to determine if a network is tightly bounded, diversified, or constricted; to find its density and clustering; and to study how the behavior of network members is affected by their positions and connections (Garton, Haythornthwaite & Wellman, 1997; Hanneman, 1998; Scott, 2000). Network researchers have developed a set of theoretical perspectives of network analysis. Some of these are (Borgatti, 2002),

- Focus on relationships between actors than the attributes of actors;
- Sense of interdependence: a molecular rather than atomistic view;
- Structure affects substantive outcomes;
- Emergent effects.

"The aim of social network analysis is to describe why people communicate individually or in groups" (Preece, 2000, p. 183), while the goals of SNA are (Dekker, 2002)

- To visualize relationships/communication between people and/or groups using diagrams;

TABLE 30.3. TAT Categories

Category
1: Questioning
The questioning category is further broken down into two types of questions:
1A Vertical Questions
These questions assume a "correct" answer exists, and that they can be answered if the right authority to supply it can be found. Example: "Does anybody know what time the library opens on Saturdays?"
1B Horizontal Questions
For these questions, there may not be only one right answer. These questions invite negotiation. Example: "Do you really think mp3 files are should become illegal, or you don't see any harm by them?"
2: Statements
This category consists of two subcategories:
2A Nonreferential Statements
These statements contain little self-revelation and usually do not invite response or dialogue and their main intent is to impart facts or information. Example: "We found that keeping content up-to-date, distribution and PC compatibility issues were causing a huge draw on Ed. Centre time."
2B Referential Statements
Referential statements are direct answers to questions. They can include comments referring to specific preceding statements. Example: "That's right; it's the 1997 issue that you want."
3: Reflections
Reflections are significant personal revelations, where the speaker expresses personal or private thoughts, judgments, opinions, or information. Example: "My personal opinion is that it shouldn't have been a penalty kick."
4: Scaffolding and Engaging
Scaffolding and engaging initiate, continue, or acknowledge interpersonal interaction. They personalize the discussion and can agree with, thank or otherwise recognize someone for their helpfulness and comments. Example, "Thanks Dave; I've been trying to figure that out for ages :)"
5: References/Authorities
Category 5 is compromised of two types:
5A Quotations, references to, paraphrases of other sources. Example, "You said, 'I'll be out of the city that day.'"
5B Citations, attributions of quotations and paraphrases. Example: "Mathew, P. (2001). A beginners' guide to mountain climbing."

- To study the factors which influence relationships and the correlations between them;
- To draw out implications of the relational data, including bottlenecks;
- To make recommendations to improve communication and workflow in an organization.

Preece (2002) and Beidernikl and Paier (2003) listed the following as the limitations of SNA:

- More theory that speaks directly to developers of online communities is needed;
- The data collected may be personal or private;
- The analysis of the data is quantitative and specific to the particular network, while common survey data are qualitative and generalize answers on the parent population.

It is also worth pointing out that network analysis is concerned about dyadic attributes between pairs of actors (like kinship, roles, and actions), while social science is concerned with monadic attributes of the actor (like age, sex, and income).

There are two approaches to SNA:

Ego-centered analysis—Focuses on the individual as opposed to the whole network, and only a random sample of network population is normally involved (Zaphiris, Zacharia, & Rajasekaran, 2003). The data collected can be analyzed using standard computer packages for statistical analysis like SAS and SPSS (Garton, Haythornthwaite, & Wellman, 1997).

Whole network analysis—The whole population of the network is surveyed and this facilitates conceptualization of the complete network (Zaphiris et al., 2003). The data collected can be analyzed using microcomputer programs like UCINET and Krackplot (Garton et al., 1997).

The following are important units of analysis and concepts of SNA (Garton et al., 1997; Wellman, 1982; Hanneman, 2001; Zaphiris et al., 2003; Wellman, 1992):

- Nodes: The actors or subjects of study;
- Relations: The strands between actors. They are characterized by content, direction, and strength;
- Ties: Connect a pair of actors by one or more relations;
- Multiplexity: The more relations in a tie, the more multiplex the tie is;
- Composition: This is derived from the social attributes of both participants;
- Range: The size and heterogeneity of the social networks;
- Centrality: Measures who is central (powerful) or isolated in networks;
- Roles: Network roles are suggested by similarities in network members' behavior;
- Density: The number of actual ties in a network compare to the total amount of ties that the network can theoretically support;
- Reachability: In order to be reachable, connections that can be traced from the source to the required actor must exit;
- Distance: The number of actors that information has to pass through to connect one actor with another in the network;
- Cliques: Subsets of actors in a network, who are more closely tied to each other than to the other actors who are not part of the subset.

Social Network Analysis is a very valuable technique when it comes to analyzing online communities as it can provide a visual presentation of the community and, more importantly, it can provide us with qualitative and quantitative measures of the dynamics of the community. The application of SNA to the analysis of online communities is further demonstrated with a case study later in this chapter.

CASE STUDIES

In this section we present two case studies that demonstrate the use of theoretical and analytical techniques for studying online communities. In the first case study, we demonstrate how the results from an "attitude toward thinking and learning" questionnaire can be combined with social network analysis to describe the dynamics of a computer-aided language-learning (CALL) online community. In the second case study, we present a theoretical activity theory model that can be used for describing interactions in online game communities.

Computer Aided Language Learning Communities

In the first case study we demonstrate a synthetic use of quantitative (SNA) and qualitative (questionnaires) methods for analyzing the interactions that take place in a Computer Aided Language Learning (CALL) course. Data was collected directly from the discussion board of the "Learn Greek Online" (LGO) course (Kypros-Net Inc., 2005).

LGO is a student-centered e-Learning course for learning Modern Greek and was built using a participatory design and distributed constructionism methodology (Zaphiris & Zacharia, 2001). In an ego-centered SNA approach, we have carried out an analysis of the discussion postings of the first 50 actors (in this case the students of the course) of LGO.

To carry out the social network analysis we used an SNA tool called "NetMiner" (Cyram, 2004), which enabled us to obtain centrality measures for our actors. The "in and out degree centrality" was measured by counting the number of interaction partners per each individual in the form of discussion threads (for example, if an individual posts a message to three other actors, then his or her out-degree centrality is three; whereas if an individual receives posts from five other actors then his or her in-degree is five).

Due to the complexity of the interactions in the LGO discussion we had to make several assumptions in our analysis:

- Posts that received zero replies were excluded from the analysis. This was necessary in order to obtain meaningful visualizations of interaction;
- Open posts were assumed to be directed to everyone who replied;
- Replies were directed to all the existing actors of the specific discussion thread unless the reply or post was specifically directed to a particular actor.

In addition to the analysis of the discussion board interactions we also collected subjective data through the form of a survey. More specifically, the students were asked to complete an Attitudes Towards Thinking and Learning Survey (ATTLS). The ATTLS uses 20 Likert scale questions to measure the extent to which a person is a "connected knower" (CK) or a "separate knower" (SK). People with higher CK scores tend to find learning more enjoyable, and are often more cooperative and congenial, and more willing to build on the ideas of others, while those with higher SK scores tend to take a more critical and argumentative stance to learning (Galotti, Clinchy, Ainsworth, Lavin, & Mansfield, 1999).

The out-degree results of the social network analysis are depicted in Fig. 30.1 in the form of a sociogram, and the in-degree results are depicted in Fig. 30.2. Each node represents one student (to protect the privacy and anonymity of our students, their names have been replaced by a student number). The position of a node in the sociogram is representative of the centrality of that actor (the more central the actor the more active). As can be seen from Fig. 30.1, students S12, S7, S4, and S30 (with out-degree scores ranging from 0.571 to 0.265) are at the centre of the sociogram and possess the highest out-degree. The same students also posses the highest in-degree scores (Fig. 30.2). This is an indication that these students are the most active members of this online learning community, posting and receiving the largest number of postings. In contrast participants in the outer circle (e.g., S8, S9, S14, etc.) are the least active with the smallest out-degree and in-degree scores (all with 0.02 out-degree scores).

In addition, a clique analysis was carried out (Fig.30. 3) and it showed that 15 different cliques (the majority of which are overlapping) of at least three actors each have been formed in this community.

As part of the ego-centered analysis for this case study we look in more detail at the results for two of our actors: S12, who is the most central actor in our SNA analysis (e.g., with the highest out-degree score), and S9, an actor with the smallest out-degree score. It is worth noting that both members joined the discussion board at around the same time.

First, through a close look at the clique data (Table 30.4) we can see that S12 is a member of 10 out of the 15 cliques, whereas S9 is not a member of any—an indication of the high interactivity of S12 versus the low interactivity of S9. In an attempt to correlate the actors' position in the SNA sociogram with their self-

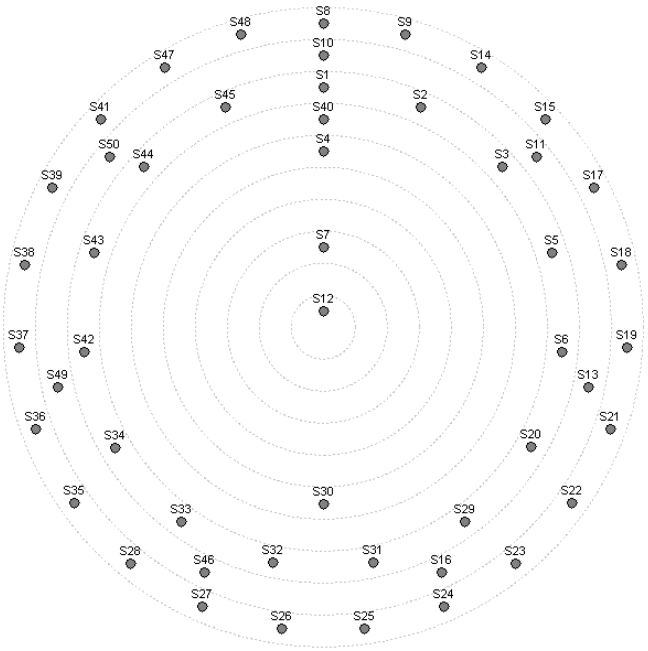


FIGURE 30.1. Out-degree analysis sociogram.

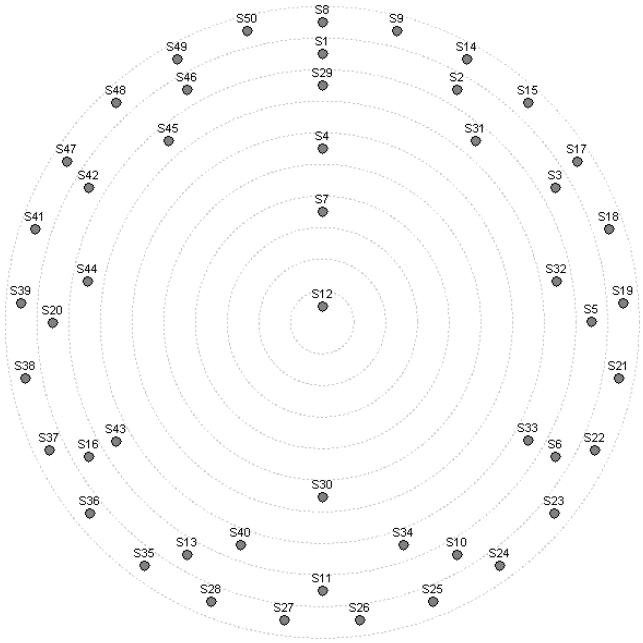


FIGURE 30.2. In-degree analysis sociogram.

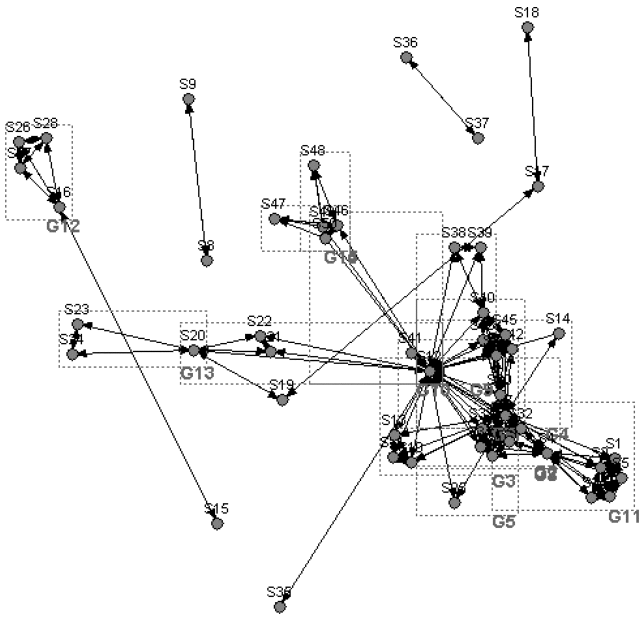


FIGURE 30.3. Clique analysis sociogram.

reported attitudes towards teaching and learning we looked more closely at the answers these two actors (S12, S9) provided to the ATTLS. Actor S12 answered all 20 questions of the ATTLS with a score of at least 3 (on a 1–5 Likert scale), whereas S9 had answers ranging from 1 to 5. The overall ATTLS score of S12 is 86 whereas that of S9 is 60. A clear dichotomy of opinions occurred on 5 of the 20 questions of the ATTLS. S12 answered all 5 of those questions with a score of 5 (strongly agree) whereas

TABLE 30.4. Clique Analysis of the LGO Discussions

Cliques	Actors
K1	S12, S7, S30, S40, S43, S44, S45
K2	S12, S7, S30, S4
K3	S12, S7, S10, S11, S13
K4	S12, S7, S14
K5	S12, S7, S25
K6	S12, S7, S41
K7	S12, S20, S21, S22
K8	S12, S29, S4, S30, S31, S32, S33, S34
K9	S12, S38, S39, S40
K10	S12, S46, S49, S50
K11	S1, S2, S3, S4, S5, S6, S7
K12	S16, S26, S27, S28
K13	S23, S20, S24
K14	S47, S46, S49, S50
K15	S48, S46, S49, S50

S9 answered them with a score of 1 (strongly disagree). More specifically, S12 strongly agreed that

1. S/He is more likely to try to understand someone else's opinion than to try to evaluate it;
2. S/He often find herself/himself arguing with the authors of books read, trying to logically figure out why they're wrong;
3. S/He finds that he/she can strengthen his/her own position through arguing with someone who disagrees with them;
4. S/He feels that the best way achieve his/her own identity is to interact with a variety of other people;
5. S/He likes playing devil's advocate, arguing the opposite of what someone is saying.

S9 strongly disagreed with all of the above statements. These are all indications that S12 is a "connected knower" (CK) whereas S9 is a "separate knower" (SK).

This case study showed that the combination of quantitative and qualitative techniques can facilitate a better and deeper understanding of online communities. SNA was found to be a very useful technique for visualizing interactions and quantifying strengths and dynamics in online communities. In combination with the ATTLS, it was possible to identify the key players of the e-Learning community. These members' roles show them to be more powerful and central in the discussions. Identifying their characteristics enables us to reinforce the community by making other participants more active in the discussion board communication. This active learning approach could, in turn, improve the pedagogical value of e-learning within these communities.

Game Communities and Activity Theoretical Analysis

The main motivation of the second case study arises from the more general area of computer game-based learning. Game-based learning has focused mainly on how the game itself can be used to facilitate learning activities but we claim that the educational opportunity in computer games stretches beyond the learning activities in the game per se. Indeed, if you observe

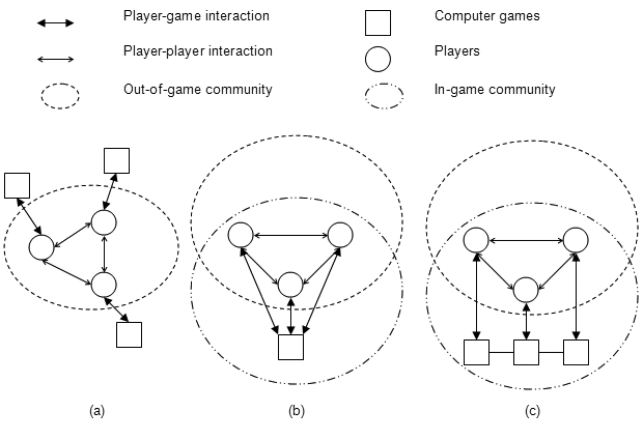
most people playing games, you will likely see them downloading guidelines from the Internet and participating in online forums to talk about the game and share strategies. In actuality, almost all game playing could be described as a social experience, and it is rare for a player to play a game alone in any meaningful sense (Kuo, 2004). This observation is even more evident in Massively Multiplayer Online Role Playing Games (MMORPGs) which has been discussed earlier in this chapter. For example, the participation in a MMORPG is constituted through language practice within the in-game community (e.g., in-game chatting and joint tasks) and out-of-game community (e.g., the creation of written game-related narratives and fan-sites). The learning is thus not embedded in the game, but it is in the community practice of those who inhabit the game.

Types of game communities. Because of the points mentioned earlier, we believe that the study on computer games should be expanded to include the entire game community. Computer game communities can be categorized into three classes which we have identified (Fig. 30. 4) (Ang, Zaphiris & Wilson, 2005) as:

Single Game-play Community—This refers to a game community formed around a single-player game. Although players of a single-player game like "The Sims 2" and "Final Fantasy VII" play the game individually, they are associated with an out-of-game community which discusses the game either virtually or physically.

Social Game-play Community—This refers to multiplayer games which are played together in the same physical location. It creates game communities at two levels: in-game and out-of-game. Occasionally, these two levels might overlap. The out-of-game interaction might be affected by issues beyond the specific game system; for example, the community starts exchanging information about another game.

Distributed Game-play Community—This is an extension of social game-play community, but it emphasizes the online multiplayer game in which multiple sessions of game are established in different geographical locations.



(a) Single game-play community (b) Social game-play community (c) Distributed game-play community

FIGURE 30.4. Types of game communities.

The study of game communities, especially out-of-game communities, from the perspective of education is still very much unexplored. We believe the potential of games in education is not limited to what is going on in the game. Educators could benefit by studying games as a social community because games are now becoming a culture that permeates the lives of many, especially the younger generation. Black (2004) has investigated the interactions among participants in a virtual community of Japanese comic fans which involve a lot of reading and writing throughout the site. She examines how the community of fans helps each its members with English language writing skills and with cross-cultural understanding. In this section we have pointed out that game communities can emerge from both single-player and multiplayer games. We believe that by further studying the social interaction in the game community, we will be able to utilize games to learn in a more fruitful way. In the next section, we apply and evaluate one of these models of game communities to a specific scenario in knowledge building using activity theory.

Activity theory. In this case study, we demonstrate how activity theory can be used to analyze an out-of-game community around a single-player game which is based on constructionist activities as proposed by Papert (1980). Papert claimed that even for adults, learning remains essentially bound to context, in which knowledge is shaped by the use of external supports. His approach helps us understand how learning is actualized when individual learners construct their own favorite artifacts or objects-to-think-with (Papert, 1980).

Although Papert's theory provided a solid framework for understanding children's, and even adults', ways of learning by designing, it does not offer a systematic framework for analyzing the construction activities within a learning community. Analyzing constructionist activities can be useful as it could help inform constructional design for educational purposes. The most significant analysis includes the learning within a community as well as the development of an individual. We are also interested in finding out how tools such as computers help learners construct artifacts and knowledge. Hence, we would like to draw from the Vygotskian naturalist approach which emphasizes human activity systems. Lev Vygotsky (1930) formulated a theoretical concept which is very different from the prevailing understanding of psychology which was dominated by behaviorism at that time. This new orientation was a model of tool-mediation and object-orientedness. He proposes the classic triangle model to demonstrate the idea of mediation:

In Fig. 30.5, the subject is the individual engaged in the mediated action, and the mediating artifact or tool could include physical artifacts and/or prior knowledge of the subject. The object is the goal/objective of the activity. Although constructionist learning relies very much on computational tools, the concept of mediation is not explicated. Fig. 30.5 shows explicitly that the relationship between the subject and the object is no longer straightforward; instead, it is mediated by the tool. For example, when building a website, the subject is working towards an objective (e.g., to add a table in the web page) using not only the computer (external tools) but also her internal understanding of how websites and computers work (internal tools).

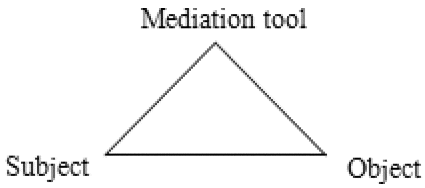


FIGURE 30.5. Vygotsky's mediation.

Leont'ev (1978) extended this notion of activity to differentiate between an individual action and a collective activity by proposing a hierarchy of activity (Table 30.5). Collective activity is connected to the object of the whole community, of which each individual subject is often not consciously aware. An individual action is connected to a conscious goal. Below the collective activity and individual action there is the level of operations that are dependent on the conditions in which the action is performed. Thus, an activity system can be analyzed at three levels: (a) the activity level which is oriented toward the object/objective and carried out by the whole community; (b) the action level which is directed at the individual goal, and (c) the operation level which is elicited by conditions and is performed unconsciously.

This hierarchy is crucial in explaining the learning process in an activity system. We would like to illustrate an example of this hierarchy in learning a foreign language (Table 30.5). The overall objective is to be able to engage in a meaningful conversation. In the beginning, the learner has to work on the grammar and the choice of words at a conscious level. When the learner has reached a higher proficiency level, these actions are transformed into operations. The learner no longer needs to select appropriate words and check grammar rules deliberately, as these have been learned thoroughly and are now operating unconsciously. The consciousness of the learner is now focused on expressing himself properly, depending on the objective of the conversation. Grammatical rules become invisible to the learner and he is only selecting appropriate goals to be achieved. Therefore, it can be inferred that activity theory treats learning as the shift from the higher level (action) to the lower level (operation) in the hierarchy.

Drawing on work by Vygotsky and Leont'ev, Engeström (Engestrom, 2001) viewed all human activities as contextualized within an interdependent activity system. Engeström added collective mediation to Vygotsky's tool mediation and presents the triangle model of activity system (Fig. 30.6).

TABLE 30.5. Hierarchy of Activity

Unit of Analysis	Stimulus	Subject	Language Learning Example
Activity	Object	Community	Engage in a meaningful conversation
Action	Goal	Individual	Sentence construction
Operation	Conditions	Unconscious	Word selections, grammar rules

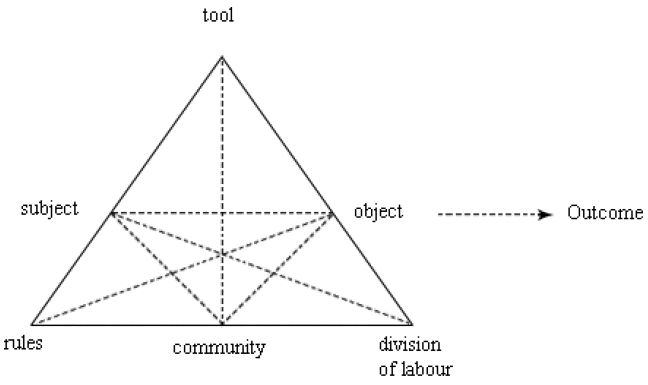


FIGURE 30.6. The triangle activity system diagram.

In the diagram, the subject is the individual or group selected as the point of view of the analysis. The object refers to the raw material or the problem space at which the activity is directed and which is transformed into outcomes with the help of external and internal tools. Tools are the concepts, physical tools, artifacts, or resources that mediate a subject's interactions with an object. The community refers to those with whom the subject shares the same general object. The division of labor (DOL) is the classification of tasks among the members on the community, while the rules are the regulations, norms, and conventions within the activity system.

Constructionist learning can be described and visualized through this activity triangle. Mediated by the tool and the community, the learner externalizes her initial stage of knowledge through object construction. The individual externalization (mediated by the tool) can be broken down into actions and operations. Actions are directed toward a personal goal and are carried out with careful deliberation. For example (Fig. 30.7), in a book-writing activity, the author (an expert word-processor user) will operate (e.g., type) the word processor at the unconscious level, and consciously act on the book (to select appropriate words, and construct meaningful sentences and paragraphs) she is writing. At a certain point, the author encounters a new condition with the word processor which she is not familiar with—for

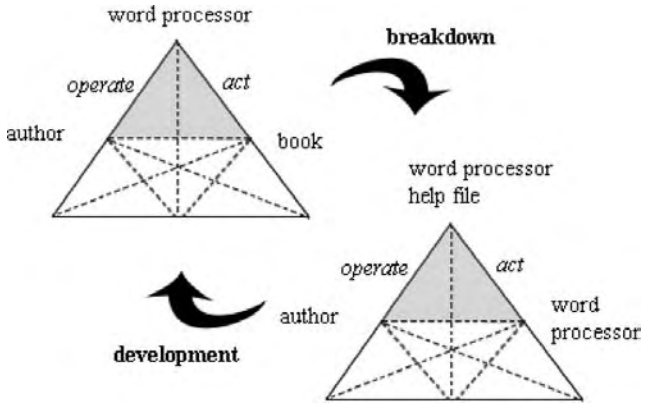


FIGURE 30.7. The transformation of individual action-operation.

example, having to insert a table into the book. Under this new condition, a breakdown is said to have happened. The conscious effort of the author is not any longer placed on the book itself but instead is now placed on the word processor (e.g., to achieve the action (insert tables), the author performs the operation (read help files)). Once the author has thoroughly learned about the table insert, she can again act on the book consciously and development is said to have happened.

Activity theoretical analysis for constructionist learning. We have conducted a study to analyze a Wiki-based game community using activity theory. The goal of this study was to demonstrate the usefulness of activity theory in researching online communities. The collaborative construction activity in a Wiki-supported community devoted to writing a game guidebook is examined. Activity theory is used as an analytical tool in order to investigate this community. In this section, experiences and challenges from the analysis are reported in order to give some insights into how activity theory can be helpful for online community research.

The analysis of online communities can be done through the lenses of various concepts associated with activity theory, such as the levels of activity hierarchy and different perspectives of the triangle model as explained in the previous section.

For instance, we can start our analysis with the most basic aspect of constructionism by simply examining the relationship between subjects and the object. Then, we can analyze Vygotsky's mediation model of activity system consisting of individual actions and tools. The analysis may also be extended to the whole community of this system to include emerging rules as well as division of labor (DOL) that mediate the community. The focus can also be placed mainly on the constructionist concept of externalizing the internal meanings onto a sharable artifact through mediation. More specifically, we can look into these (and other) aspects:

- Subject-object: What are the constructionist actions that act on the object and transform the object into outcome?
- Subject-tool-object: How do actions shift to operations and vice versa? How do tools mediate individual actions and operations? What is the nature of the mediating tools? How do they support knowledge building?
- Subject-rules-object and subject-DOL-object: What is the nature of implicit and explicit rules? How is DOL manifest in the community? How do rules and DOL support knowledge building?

Apart from its focus on both individual and collective development through action-operation transformation, activity theory also helps analyze the tools and capture the rules and the division of labor (DOL) which mediate the actions. These uses must be further explained in order to differentiate individual mediation and collective mediation. Taking the example of our study on game community, individual mediation places its emphasis on how a user uses the tool to write the game guide, without taking into account how other users act in the community. In other words, it is about the affordance of the tool to support what an individual can do.

Collective mediation is about the community, which consists of two major components: rules and DOL. Rules define what can be done and cannot be done in a community. This should not be confused with the affordance of the tool. The tool might afford certain actions such as writing in abusive language, but the rules might want to ban this action. DOL is self-explained: how the workload is divided among many users in a community.

Activity theory appears to be a promising framework as it gives an analytical lens on analyzing and interpreting the data. Activity theory provides different perspectives of analysis, as it casts different light on the collected data as researchers can examine it from many perspectives by focusing on different sub-triangles of the activity system diagram. It also helps us examine learning process—how learning occurs individually and collectively through the transformation of hierarchy of activity from action to operation. Furthermore, both individual and collective aspects are given equal importance. Activity theory informs the development of the whole community as well as the individual's development. It explains how individual development contributes to the community growth, and vice versa.

Based on our study, individual actions help sharpen the mediation tool, while collective actions bring about new rules or refine existing rules that mediate the collective action. In short, activity theory is useful to analyze the community in the following ways:

- It helps understanding of the individual mediation process: subject-tool-object;
- It clearly presents the collective mediation process: subject-community-object;
- It reveals the emerging rules and division of labor in the community.

In a Wiki space, knowledge is socially constructed; it is created individually with tools and negotiated and agreed to within a community based on emerging rules and division of labor (DOL). It starts as a single unit of information (a page in this context) and grows organically and evolves into a complex and well-structured set of knowledge. From our findings, we induce what contributes to the development of the community:

- Users share some historical backgrounds: they already share some of the tools/rules before joining the community, they also share the interest on the same game;
- Users share the same object (goal) which is to build a game guide book;
- A user's individual action: this goal-oriented individual action triggers negotiations that leads to the growth of the space;
- The community's agreement on the object: not only share the same object, the community must be able to negotiate and agree on the object;
- Tools that support these actions and negotiations;
- Emerging rules that coordinate the activity;
- DOL that divide the responsibilities.

The evolution of a knowledge-building community needs more than a group of devoted users who share the same ob-

ject. It involves negotiation and agreement among the users on the object. Although every user tends to act toward her or his own goal, it takes the compromise of the entire community to agree on the object.

Activity theory helps online community researchers identify design issues at two aspects: the software's application as tools as well as the social interaction within the community around the tools. It also reveals the development of the community-building process from the individual and collective level, through the shift of activity hierarchy. We thus believe that analyzing online communities with activity theory will yield fruitful results and give insights on online community design.

Finally, we must reiterate the fact that activity theory itself is not limited to what is presented in the triangle diagram as proposed by Engeström. Although his model is tremendously useful, it overlooks several significant concepts of activity theory. One major limitation is the static representation of activity theory. The triangle diagram represents only a snapshot of a particular time, thus making it hard to analyze the activity across time. It is understood that Engeström's model is intended to be open so that it can be used in various domains but this has been proven to pose serious difficulty among the practitioners, as some researchers have started to operationalize it so that it is more practical in day-to-day methodology (Korpela, Soriyan, & Olufokunbi, 2000; Barab, Hay, & Yamagata-Lynch, 2001; Mwanza, 2002). Hopefully with the increasing attention drawn by activity theory, the theory will be expanded and operationalized to fit the purpose of HCI research in general and online community in particular.

DISCUSSION AND CONCLUSIONS

This chapter looked at the definitions of computer mediated communication and online communities and pointed out the multi-disciplinary nature of the definitions and the way online communities are being analyzed and studied. We introduced the different types of CMC and online communities and put special attention to Wikis and game-based communities. These two relatively new types of communities exhibit new modes of interaction that are worth studying further. We provided an overview of some of the most commonly used techniques and theoretical frameworks for analyzing online communities. Then, we used two case studies to demonstrate the use of those techniques. Although both of our case studies focus on e-learning communities, communities have been used in a multitude of disciplines. Empathic online communities, for example, seem to be gaining a lot of momentum, with people supporting each other. Online communities are widely used for general entertainment and current affairs discussions as well. Professional communities (e.g., business, art, industry specific) are also being formed within the online environment.

The study of online communities is flourishing, primarily due to the increasing popularity of online services and tools that provide the construction of such networks of users. The study of such complex communities requires the use of a synthesis of methodologies and theoretical foundations. In our first case study, we demonstrated how Social Network Analysis can be used to model and visualize online community interactions; in the second case study, the theoretical foundations based on activity theory were applied to the domain of game-based communities.

At the beginning of the Internet technology, online communities were solely used for synchronous (often just chat) or asynchronous (most commonly discussion board) textual interaction. This is no longer the case, as people are interacting in online communities using new and more innovative interaction paradigms. Game-based communities, for instance, allow users to represent themselves (using 3D avatars) visually in virtual environments which are often depicted as a fantastic and unrealistic world. They can navigate, change, or even create the virtual world (and thus the context of the community) they interact with. As such, the traditional boundary between author and reader is distorted as the designer (authors) is not the one and only who determines what the system is like. Rather, the participants (readers) are coconstructing the virtual world as they are not anymore using the tool to communicate; they are creating and interacting with a virtual environment through which they can meet, socialize, and work with others. Preece (2000) was the first to identify and stress this important social dimension of online communities. Now, this online sociability is becoming more mature and more central to the online communities with which we interact.

This new paradigm of interaction poses new challenges for researchers and practitioners. The importance of appropriate representation of emotions and other social and cultural cues in online communities is now becoming even more important. With textual interfaces there was an attempt to represent these social cues using emoticons. How can this, for example, be transferred to the domain of virtual game communities? Do we want our avatars to behave like us or do we want them to have some alternate and illusionary identities with extraordinary abilities or unusual behaviors? How can we come up with new interfaces and new interaction paradigms that can facilitate this better, in order to cope with the new demands from the users in such online communities?

A second area which is gaining popularity is the research of online communities, or Internet research. Content analysis and query-based techniques were sufficient if what we wanted was a first good impression of the interactions taking place in an online textual community. This is not anymore the case. Studying the sociability of game-based communities, for example, requires the synthesis of more techniques. We might want to immerse ourselves in that community, engage in long-term ethnographic studies of its environment, and get the first-hand experiences of what happens in it. We might have to employ social-cultural theoretical frameworks (e.g., Activity Theory) to get a better understanding of the ways people behave differently from the real world in these inherently social environments. We might have to quantify, using Social Network Analysis for example, the dynamics of the networks and subgroups that evolve around these communities.

The possibility of online communities is unlimited with the emergence of more mature and imaginative virtual worlds. Only by treating them in equality with their physical counterparts which encapsulate the practices of economy, politics, ideology, and everyday life, can we research and study them intellectually.

References

Ang, C. S., Zaphiris, P., & Wilson, S. (2005). Social interaction in game communities and second language learning. *The 19th British HCI Group Annual Conference*, Edinburgh, UK

Archer, W., Garrison, R. D., Anderson, T., & Rourke, L. (2001). A framework for analyzing critical thinking in computer conferences. *European Conference on Computer-Supported Collaborative Learning*, Maastricht, Netherlands.

Aronsson, L. (2002). Operation of a large scale, general purpose Wiki website. Experience from susning.nu's first nine months in service. In J. Á. Carvalho, A. Hübler, & A. A. Baptista (Eds.), *Proceedings of the 6th International ICCC/IFIP Conference on Electronic Publishing* (pp. 27–37). Karlovy Vary, Czech Republic: Verlag für Wissenschaft und Forschung.

Barab, S. A., Hay, K. E., & Yamagata-Lynch, L. C. (2001). Constructing networks of action-relevant episodes: An In Situ research methodology. *The Journal of Learning Sciences, 10*(1 & 2), 63–112.

Bates, A. W. (1995). *Technology, open learning and distance education*. London: Routledge.

Beidernikl, G., & Paier, D. (2003). Network analysis as a tool for assessing employment policy. *Proceedings of the Evidence-Based Policies and Indicator Systems Conference*, London.

Black, R. W. (2004). *Anime-inspired affiliation: An ethnographic inquiry into the literacy and social practices of English language learners writing in the fan-fiction community.* Paper presented at 2004 meeting of American Educational Research Association, San Diego, CA.

Blomkvist, S. (2002). *Persona—an overview, Uppsala University.* Retrieved November 22, 2004, from http://www.it.uu.se/edu/course/homepage/hcinet/ht04/library/docs/Persona-overview.pdf

Borgatti, S. (2000). *What is social network analysis*. Retrieved November 9, 2004, from http://www.analytictech.com/networks/whatis.htm

Burge, E. L. & Roberts, J. M. (1993). *Classrooms with a difference: A practical guide to the use of conferencing technologies*. Ontario: University of Toronto Press.

CAP, University of Warwick (2004). *E-Guide: Using computer mediated communication in learning and teaching*. Retrieved November 8, 2004, from http://www2.warwick.ac.uk/services/cap/resources/eguides/cmc/cmclearning/

Castronova, E. (2001, December). *Virtual worlds: A first-hand account of market and society on the cyberian frontier*. CESifo Working Paper Series No. 618. Fullerton: Center for Economic Studies and Ifo Institute for Economic Research, California State University.

Choi, J. H., & Danowski, J. (2002). Cultural communities on the net—Global village or global metropolis: A network analysis of Usenet newsgroups. *Journal of Computer-Mediated Communication, 7*(3). Retrieved March 8, 2007, from http://JCMC.indiana.edu/vol7/issue3/chai.html

Cook, D., & Ralston, J., (2003). Sharpening the focus: Methodological issues in analyzing on-line conferences. *Technology, Pedagogy and Education, 12*(3), 361–376.

Cooper, A. (1999). *The Inmates are running the asylum*. Indianapolis, IN: SAMS, a division of Macmillan Computer Publishing.

Cyram. (2004). *Netminer for windows*. Retrieved March 8, 2007, from http://netminer.com

December, J. (1997). Notes on defining of computer-mediated communication. *Computer-Mediated Communication Magazine, 3*(1). Retrieved March 8, 2007, from http://www.december.com/cmc/may/1997/jan/december.html

Dekker, A. H. (2002, January–February). A category-theoretic approach to social network analysis. *Proceedings of Computing: The Australasian Theory Symposium* (*CATS*). Melbourne, Australia.

Ducheneaut N., Moore R. J., & Nickell, E. (2004). *Designing for sociability in massively multiplayer games: An examination of the "Third Places" of SWG*. Other Players Conference, Denmark.

Electronic Art (2005). *Ultima online*. Retrieved November 8, 2005, from http://www.uo.com/

Emigh, W., & Herring, S. C. (2005). Collaborative authoring on the web: A genre analysis of online encyclopedias. *Proceedings of the 38th Hawai'i International Conference on System Sciences* (HICSS-38). Los Alamitos, CA: IEEE Press.

Engestrom, Y. (2001). Expansive learning at work: Toward an activity theoretical reconceptualisation. *Journal of Education and Work, 14*(1), 133–156.

Fahy, P. J. (2003). Indicators of support in online interaction. *International Review of Research in Open and Distance Learning, 4*(1). Retrieved March 8, 2007, from http://www.irrodl.org/index.php/irrodl/article/view/129/209

Fahy, P. J., Crawford, G., & Ally, M., (2001). Patterns of interaction in a computer conference transcript. *International Review of Research in Open and Distance Learning, 2*(1). Retrieved 8 March 2007, from http://www.irrodl.org/irrodl/article/view/36/73

Ferris, P. (1997). What is CMC? An overview of scholarly definitions. *Computer-Mediated Communication Magazine, 4*(1). Retrieved March 8, 2007, from http:www.december.com/cmc/may/1997/jun/december.html

Fortner, R. S. (1993). *International communication: History, conflict, and control of the global metropolis*. Belmont, CA: Wadsworth.

Galotti, K. M., Clinchy, B. M., Ainsworth, K., Lavin, B., & Mansfield, A. F. (1999). A new way of assessing ways of knowing: The attitudes towards thinking and learning survey (ATTLS). *Sex Roles, 40*(9 & 10), 745–766.

Garton, L., Haythornthwaite, C., & Wellman, B. (1997). Studying On-line Social Networks. In Jones, S. (Ed.), *Doing internet research*. Thousand Oaks, CA: Sage.

Gunawardena, C., Lowe, C., & Anderson, T. (1997). Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. *Journal of Educational Computing Research, 17*(4), 397–431.

Halvorsen, R. F. (2005). *TheStateOfWiki*. Retrieved September 22, 2005, from http://heim.ifi.uio.no/~aurilla/web/rune.pdf

Hanneman, R. A. (2001). *Introduction to social network methods.* Retrieved November 9, 2004, from http://faculty.ucr.edu/~hanneman/SOC157/TEXT/TextIndex.html

Heeren, E. (1996). *Technology support for collaborative distance learning*. Doctoral dissertation, University of Twente, Enschede.

Henri, F. (1992). Computer conferencing and content analysis. In A. R. Kaye (Ed.), *Collaborative learning through computer conferencing: The Najaden Papers* (pp. 117–136). Berlin: Springer-Verlag.

Jones G. B. (2003). Blogs and Wikis: Environments for on-line collaboration. *Language Learning & Technology*, *7*(2), 12–16.

Jones, S. (1995). Computer-mediated communication and community. *Computer-Mediated Communication Magazine, 2*(3).Retrieved March 8, 2007, from http://www.december.com/cmc/may/pms/mar/jones.html

Kelly, R. V. (2004). *Massively multiplayer online role-playing games: The people, the addiction and the playing experience*. McFarland & Company, U.S.

Kittowski, F. F., & Köhler, A. (2002). Knowledge creating communities in the context of work processes. *ACM SIGGROUP Bulletin, 23*(3), 8–13.

Kolbert, E. (2001, May 28). Pimps and dragons: How an online world survived a social breakdown. *The New Yorker*. Retrieved February 15, 2005, from http://www.newyorker.com/fact/content/?010528fa_FACT

Korpela, M., Soriyan, H. A., & Olufokunbi, K. C. (2000). Activity analysis as a method for information systems development. *Scandinavian Journal of Information Systems, 12*, 191–210.

Korzenny, F. (1978). A theory of electronic propinquity: Mediated communication in organizations. *Communication Research, 5*, 3–23.

Krebs, V. (2004). *An introduction to social network analysis*. Retrieved November 9, 2004, from http://www.orgnet.com/sna.html

Kuo, J. (2004). *Online video games in mental health*. Paper presented at the annual meeting of the American Psychiatry Association, New York.

Kypros-Net Inc. (2005). *The world of Cyprus*. Retrieved March 8, 2007, from http://kypros.org

Leont'ev, A. N. (1978). *Activity, consciousness, and personality*. New York: Prentice-Hall.

Madden, M., & Rainie, L. (2003). Pew Internet & American Life Project Surveys. Pew Internet & American Life Project, Washington, DC.

Mader, S. (2004). OpenSpectrum: A Wiki-based learning tool for Spectroscopy that anyone can edit. *Winter 2005 CONGCHEM: Trends and New Ideas in Chemical Education*, virtual conference. Retrieved March 8, 2007, from http://www.ched-ccce.org/confchem/2005/a/index.html

Mason, R. (1991). Analyzing computer conferencing interactions, *Computers in Adult Education and Training, 2*(3), 161–173.

McLoughlin, C. (1996, January 21–25). A learning conversation: Dynamics, collaboration and learning in computer mediated communication. In C. McBeath & R. Atkinson (Eds.), *Proceedings of the Third International Interactive Multimedia Symposium*, Promaco Conventions, Perth, Western Australia, 267–273.

McLuhan, M. (1964). *Understanding media: The extension of man*. New York: McGraw Hill.

Metcalfe, B. (1992). *Internet fogies to reminisce and argue at Interop Conference* (p. 45). InfoWorld.

Mwanza, D. (2002). *Towards an activity-oriented method for HCI research and practice*. PhD Thesis, Open University.

NCsoft (2005). *Lineage*. Retrieved November 8, 2005, from http://www.lineage.com/

NUA Internet Surveys (2004). Retrieved October 20, 2004, from http://www.nua.ie/surveys/how_many_online/index.html

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Preece, J. (2000). *Online Communities: Designing usability, supporting sociability*. Chichester, UK: John Wiley and Sons.

Preece, J., & Maloney-Krichmar, D. (2003) Online communities: Focusing on sociability and usability. In J. Jacko & A. Sears (Eds.), *Handbook of Human-Computer Interaction* (pp. 596–620). Mawhah, NJ: Lawrence Erlbaum Associates.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: Beyond human-computer interaction*. New York: John Wiley & Sons.

Reiser, R. A., & Gagné, R. M. (1983). *Selecting media for instruction*. Englewood Cliffs, NJ: Educational Technology Publications.

Rheingold, H. (1993). *The virtual community: Homesteading on the electonic frontier*. Reading, MA: Addison-Wesley.

Rice, R. (1994). Network analysis and computer mediated communication systems. In S. W. J. Galaskiewkz (Ed.), *Advances in social network analysis.* Newbury Park, CA: Sage.

Rice, R. E., Grant, A. E., Schmitz, J., & Torobin, J. (1990). Individual and network influences on the adoption and perceived outcomes of electronic messaging. *Social Networks, 12*, 17–55.

ScotCIT. (2003). Enabling large-scale institutional implementation of communications and information technology (ELICIT). *Using Computer Mediated Conferencing*. Retrieved November 2, 2004, from http://www.elicit.scotcit.ac.uk/modules/cmc1/welcome.htm

Scott, J. (2000). *Social network analysis: A handbook* (2nd ed.). London: Sage.

Sony (2005). *Star Wars Galaxies*. Retrieved November 8, 2005, from http://starwarsgalaxies.station.sony.com/

Squire, K., & Steinkeuhler, C. (in press): Generating cyberculture/s: The case of Star Wars Galaxies. In D. Gibbs, & K. L. Krause (Eds.), *Cyberlines: Languages and cultures of the Internet* (2nd ed.). Albert Park, Australia: James Nicholas Publishers.

Suler, J. (2004). *The final showdown between in-person and cyberspace relationships*. Retrieved November 3, 2004, from http://www1.rider.edu/~suler/psycyber/showdown.html

Taylor, T. L. (2002). Whose game is this anyway?: Negotiating corporate ownership in a virtual world. *Proceedings of the CGDC Conference*, 227–242. Tampere, Finland

U.S. Census Bureau (2004). *Global population profile 2002*. Retrieved October 20, 2004, from http://www.census.gov/ipc/www/wp02.html

Usability Net. (2003). *UsabilityNet* (2003) Interviews. Retrieved December 3, 2004, from http://www.usabilitynet.org/tools/interviews.html

Vaknin, S. (2002). TrendSiters: Games people play. *Electronic Book Web.* Retrieved February 23, 2002, from http://12.108.175.91/ebookweb

Viégas, F. B., Wattenberg, M., & Dave, K. (2004). Studying cooperation and conflict between authors with history flow visualizations. In E. Dykstra-Erickson & M. Tscheligi (Eds.), *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems*, (pp. 575–582). Vienna, Austria. ACM Press

Vygotsky, L. (1930). *Mind and society*. Cambridge, MA: Harvard University Press.

Wallace, P. (1999). *The psychology of the Internet*. Cambridge, UK: Cambridge University Press.

Wang, H. C., Lu, C. H., Yang, J. Y., Hu, H. W., Chiou, G. F., Chiang, Y. T., et al. (2005). An empirical exploration of using Wiki in an English as a second language course. *Proceedings of 5th IEEE International Conference on Advanced Learning Technologies* (*ICALT 2005*), Kaohsiung, Taiwan, 155–157.

Wellman, B. (1982). Studying personal communities. In P. M. N. Lin (Ed.), *Social structure and network analysis* (pp. 61–80). Beverly Hills, CA: Sage.

Wellman, B. (1992). Which types of ties and networks give what kinds of social support? *Advances in Group Processes, 9*, 207–235.

Wikipedia (2005). *Wikipedia: The free encyclopedia*. Retrieved October 1, 2005, from http://en.wikipedia.org/wiki/Wikipedia

Yee, N. (2005, May). The demographics and derived experiences of users of massively multi-user online graphical environments. Paper presented at The 55th Internal Communication Association Annual Conference, New York.

Zaphiris, P., & Zacharia, G. (2001). Design methodology of an online Greek language course. *Ext. Abstracts CHI 2001 conference*. ACM Press.

Zaphiris, P., Zacharia, G., & Rajasekaran, M. (2003). Distributed constructionism through participatory design. In Ghaoui, C. (Ed.) *E-education applications: Human factors and innovative approaches* (pp. 164–179). London, UK: Idea Group.

Zhu, E. (1996). Meaning negotiation, knowledge construction, and mentoring in a distance learning course. In *Proceedings of Selected Research and Development Presentations at the 1996 National Convention of the Association for Educational Communications and Technology*, Indianapolis, IN.

·31·

VIRTUAL ENVIRONMENTS

Kay M. Stanney
University of Central Florida

Joseph V. Cohn
Naval Research Laboratory

Introduction **622**
System Requirements **622**
Hardware Requirements 623
Multimodal I/Os 623
Tracking systems 624
Interaction techniques 625
Augmented cognition techniques 625
Software Requirements 625
Modeling 625
Autonomous agents 626
Networks 626
Design and Implementation Strategies **626**
Cognitive Aspects 626
Multimodal interaction design 626
Perceptual illusions 627
Navigation and wayfinding 628
Content Development 628
Products Liability 628
Usage Protocols 628
Health and Safety Issues **629**
Cybersickness, Adaptation, and Aftereffects 629
Social Impact 630
Virtual Environment Usability Engineering **630**
Usability Techniques 630
Sense of Presence 630
Virtual Environment Ergonomics 631
Application Domains **631**
VE as a Selection and Training Tool 631
VE as Entertainment Tool 633
VE as a Medical Tool 634
Psychological/cognitive rehabilitation applications 634
Motor rehabilitation applications 634
VE as a System Acquisition Tool 634
Conclusions **635**
Acknowledgments **635**
References **635**

INTRODUCTION

Interactive computing has evolved through the years from cryptic command-based interfaces to a collection of task-based applications to ecologically valid immersive environments, each advance dissolving more of the barrier between users and their desired actions. To many, with virtual environments (VEs) we have reached the panacea in interactive computing. Such environments immerse users in realistic settings, allowing them to engage in an intuitive and intimate manner with their digital universe. Such capability affords the opportunity to "learn by doing," "train like we fight," and "involve me and I understand." While early VEs were low in resolution and sluggish in responsiveness, advances in display and tracking technology have largely resolved these issues such that today VE applications abound (Davies, 2002). Yet, considerable HCI research and development is required to resolve lingering issues, such as cybersickness and usability, if VE technology is to be openly embraced by users. This chapter reviews the current state-of-the-art in VE technology, provides design and implementation strategies, discusses health and safety concerns and potential countermeasures, and presents the latest in VE usability engineering approaches. Current efforts in a number of application domains are reviewed. The chapter should enable readers to better specify design and implementation requirements for VE applications and prepare them to use this advancing technology in a manner that minimizes health and safety concerns.

SYSTEM REQUIREMENTS

Virtual environments provide multimodal computer generated experiences, which are driven by the hardware and software used to generate a virtual world (see Fig. 31.1). Hardware interfaces consist primarily of the

- Interface devices used to present multimodal information and sense the VE;
- Tracking devices used to identify head and limb position and orientation;
- Interaction techniques that allow users to navigate through and interact with the virtual world.

Software interfaces include the

- Modeling software used to generate VEs;
- Autonomous agents that inhabit VEs;
- Communication networks used to support multi-user virtual environments.

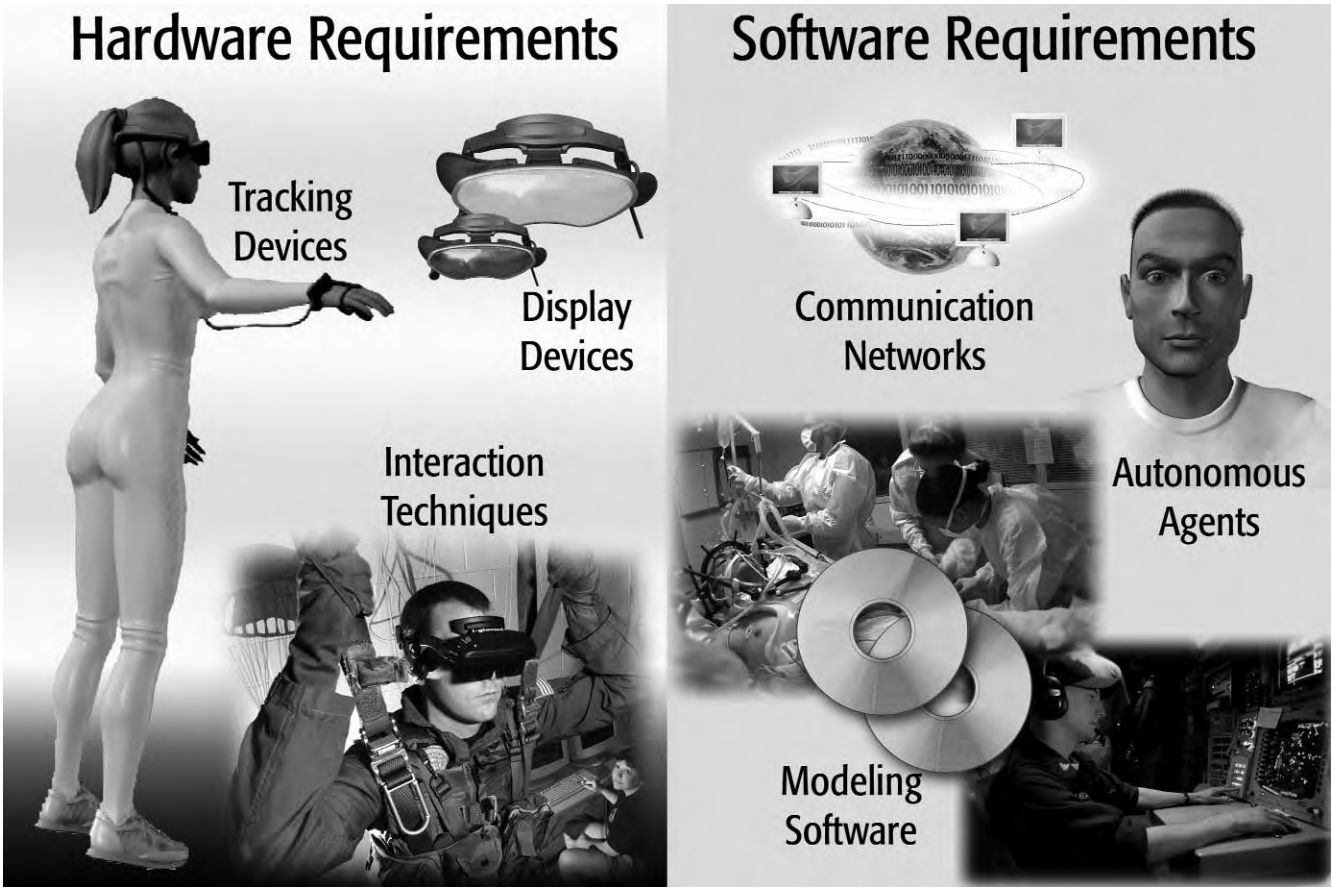


FIGURE 31.1. Hardware and software requirements for virtual environment generation.

Hardware Requirements

Virtual environments require very large physical memories, high-speed processors, high-bandwidth mass-storage capacity, and high-speed interface ports for interaction devices (Durlach & Mavor, 1995). These requirements are easily met by today's high-speed, high-bandwidth computing systems, many of which have surpassed the Gigahertz barrier (e.g., AMD's Athlon, Intel's Pentium IV and Itanium, Sun's UltraSPARC IV, IBM's Power5). The future looks even brighter, with promises of massive parallelism in quantum computing, which will allow tomorrow's computing systems to be exponentially faster than their ancestors. With the rapidly advancing ability to generate complex and large-scale virtual worlds, hardware advances in multimodal input/output (I/O) devices, tracking systems, and interaction techniques are needed to support generation of increasingly engaging virtual worlds. In addition, the coupling of augmented cognition and VE technologies can lead to substantial gains in the ability to evaluate their effectiveness.

Multimodal I/Os. To present a multimodal VE (see chapter 14), multiple devices are used to present information to VE users. Stanney and Zyda (2002) suggested that the single largest challenge in presenting multimodal VE applications is the advances required in peripheral computer connections. Currently the serial ports used to connect position trackers and other peripheral devices are typically those designed for character input rather than high-speed data transfer. This creates an input port speed problem that must be resolved if multimodal VE systems are to effectively present information to users.

In terms of peripheral devices, the one that has received the greatest attention, both in hype and disdain, is almost certainly the head-mounted display (HMD). One benefit of HMDs is their compact size, as an HMD when coupled with a head tracker can be used to provide a similar visual experience as a multitude of bulky displays associated with Spatially Immersive Displays (SIDs)and desktop solutions. There are three main types of HMDs: monocular (e.g., one display source), biocular (e.g., two displays with separate display and optics paths that show one image), and binocular (e.g., stereoscopic viewing via two image generators) (Vince, 2004). HMDs generally use CRT or LCD image sources, with CRTs typically providing higher resolution, although at the cost of increased weight. When coupled with tracking devices, HMDs can be used to present 3D visual scenes that are updated as a user moves his or her head about a virtual world. Although this often provides an engaging experience, due to poor optics, sensorial mismatches, and slow update rates, these devices are also often associated with adverse effects such as eyestrain and nausea (May & Badcock, 2002). In addition, while HMDs have come down substantially in weight, rendering them more suitable for extended wear, they are still hindered by cumbersome designs, obstructive tethers, suboptimal resolution, and insufficient fields of view. These shortcomings may be the reason behind why, in a recent review of HMD devices, more than a third had been discontinued by their manufacturers (Bungert, 2005). Of the HMDs available, there are several low- to mid-cost models, which are relatively lightweight and provide a horizontal field-of-view and resolution far exceeding predecessor systems.

Low-technology stereo viewing VE display options include anaglyph methods, where a viewer wears glasses with distinct color polarized filters; parallel or cross-eyed methods, in which right and left images are displayed adjacently (parallel or crossed), requiring the viewer to actively fuse the separate images into one stereo image; parallax stereogram methods, in which an image is made by interleaving columns of two images from a left- and right-eye perspective image of a 3D scene; polarization methods, in which the images for the left and right eyes are projected on a plane through two orthogonal linearly polarizing filters (e.g., the right image is polarized horizontally; the left is polarized vertically) and glasses with polarization filters are donned to see the 3D effect; Pulfrich methods, in which an image of a scene moves sideways across the viewer's field of view and one eye is covered by a dark filter so that the darkened image reaches the brain later causing stereo disparity; and shutter glass methods in which images for the right and left eyes are displayed in quick alternating sequence and special shutter glasses are worn that "close" the right or left eye at the correct time (Halle, 1997; Vince, 2004). All of these low-technology solutions are limited in terms of their resolution, the maximum number of views that they can display, and clunky implementation; they can also be associated with pseudoscopic images (e.g., the depth of an object can appear to flip inside out).

Other options in visual displays include Spatially Immersive Displays (SIDs) (e.g., displays that surround viewers physically with panoramic large field-of-view imagery generally projected via fixed front or rear projection display units; Majumder, 2003), desktop stereo displays, and volumetric displays that fill a volume of space with a "floating" image (Halle, 1997). Examples of SIDs include the Cave Automated Virtual Environment (CAVE) (Cruz-Neira, Sandin, & DeFanti, 1993), ImmersaDesk, PowerWall, Infinity Wall, and VisionDome (Majumder, 1999). Issues with SIDs include a stereo view that is correct for only one or a few viewers, noticeable overlaps between adjacent projections, and image warp on curved screens. Desktop display systems have advantages over SIDs because they are smaller, easier to configure in terms of mounting cameras and microphones, easier to integrate with gesture and haptic devices, and more readily provide access to conventional interaction devices, such as mice, joysticks, and keyboards. Issues with such displays include stereo that is only accurate for one viewer and a limited-display volume. Volumetric displays are advantageous because imagery can be seen from a multitude of viewing angles, generally without the need for goggles. A recent example of this type of display is the Perspecta Spatial 3D, which projects images onto a rotating flat screen inside a glass dome (Favalora et al., 2003). Issues with volumetric displays include low resolution and the tendency for transparent images to lose interposition cues. Also, view-independent shading of objects is not possible with volumetric displays, and current solutions do not exhibit arbitrary occlusion by interposition of objects (Halle, 1997).

The way of the future seems to be wearable computer displays (e.g., Microvision, MicroOptical), which can incorporate miniature LCDs directly into conventional eyeglasses (Tredennick & Shimamoto, 2005). If designed effectively, these devices should eliminate the tethers and awkwardness of current designs, while enlarging the field of view and enhancing resolution.

Stanney and Zyda (2002, p. 3) suggested that "with advances in wireless and laser technologies and miniaturization of LCDs, during the next decade visual display technology should realize the substantial gains necessary to provide high fidelity virtual imagery in a lightweight non-cumbersome manner."

When virtual environments provide audio (see chapter 9), the interactive experience is generally greatly enhanced (Shilling & Shinn-Cunningham, 2002). Audio can be presented via spatialized or nonspatialized displays. Just as stereo visual displays are a defining factor for VE systems, so are "interactive" spatialized audio displays (e.g., those with "on-the-fly" positioning of sounds). VRSonic's SoundScape3D (http://www.vrsonic.com/), Aureal's A3D Interactive (http://www.a3d.com/), and AuSIM3D (http://ausim3d.com/) are examples of positional 3D audio technology. Recently there have been promising developments in new sound modeling paradigms (e.g., VRSonic's ViBe technology) and sound design principles that will hopefully lead to a new generation of tools for designing effective spatial-audio environments (Fouad, 2004; Fouad & Ballas, 2000; Hahn, Fouad, Gritz, & Lee, 1998).

Developers must decide if sounds should be presented via headphones or speakers. For nonspatialized audio, most audio characteristics (e.g., timbre, relative volume), are generally considered to be equivalent whether projected via headphones or speakers. This is not so for spatialized audio, in which the information must generally be projected through headphones in order to exploit subtle nuances. However, spatialized audio headphones tend to be more expensive than speakers, and they also tend to require customization to an individual listener for optimum effect. Speaker systems have a larger footprint, are sensitive to room acoustics, and the listener must be placed in the "sweet spot" (e.g., ideal listening position) for the proper 3D audio effect to be realized (Shilling & Shinn-Cunningham, 2002). With true 3D audio, a sound can be placed in any location, right or left, up or down, near or far via the use of a head-related transfer function (HRTF) to represent the manner in which sound sources change as a listener moves his or her head (Begault, 1994; Butler, 1987; Cohen, 1992). Recent advances have allowed for the development of personalized HRTF functions, however, currently, these functions still require a significant amount of calibration time and fail to provide adequate cues for front-to-back or up-down localization (Crystal Rivers Engineering, 1995). Ideally, a more generalized HRTF could be developed that would be applicable to a multitude of users. This may be possible by using a best-fit HRTF selection process in which one finds the nearest matching HRTF in a database of candidate HRTFs by comparing the physiological characteristics of stored HRTFs to those of a target user (Algazi, Duda, Thompson, & Avendano, 2001).

While not as commonly incorporated into VEs as visual and auditory interfaces, haptic devices (see chapter 10) can be used to enhance aspects of touch and movement of the hand or body segments while interacting with a virtual environment. Specifically, haptic displays can be used to support two-way communications between humans and interactive systems, enabling bidirectional interaction between a user and his or her surroundings (Hale & Stanney, 2004). In general, haptic displays are effective at alerting people to critical tasks (e.g., warning), providing a spatial frame of reference within one's personal space, and supporting hand-eye coordination tasks. Texture cues, such as those conveyed via vibrations or varying pressures, are effective as simple alerts and may speed reaction time and aid performance in degraded visual conditions (Akamatsu, 1994; Biggs & Srinivasan, 2002; Massimino & Sheridan, 1993; Mulgund, Stokes, Turieo, & Devine, 2002). Kinesthetic devices are advantageous when tasks involve hand-eye coordination (e.g., object manipulation), where haptic sensing and feedback are key to performance. Currently available haptic-interaction devices include static displays (e.g., convey deformability or Braille); vibrotactile, electrotactile, and pneumatic displays (e.g., convey tactile sensations such as surface texture and geometry, surface slip, surface temperature); force feedback systems (e.g., convey object position and movement distances); and exoskeleton systems (e.g., enhance object interaction and weight discrimination) (Hale & Stanney, 2004). Advances in this area are worth pursuing because incorporating haptic feedback in VEs can substantially enhance performance (Burdea, 1996).

The "vestibular system can be exploited to create, prevent, or modify acceleration perceptions" in virtual environments (Lawson, Sides, & Hickinbotham, 2002, p. 137). For example, by simulating acceleration cues, a person can be psychologically transported from his or her veridical location, such as sitting in a chair in front of a computer, to a simulated location, such as the cockpit of a moving airplane. While vestibular cues can be stimulated via many different techniques in VEs, three of the most promising methods are physical motion of the user (e.g., motion platforms), wide field-of-view visual displays that induce vection (e.g., an illusion of self-motion), and locomotion devices that induce illusions of self-motion without physical displacement of the user through space (e.g., walking in place, treadmills, pedaling, foot platforms) (Hettinger, 2002; Hollerbach, 2002; Lawson et al., 2002). Of these options, motion platforms are probably the most advanced. Motion platforms are generally characterized via their range of motion/degrees of freedom and actuator type (Isdale, 2000). In terms of range of motion, motion platforms can move a person in many combinations of translational (e.g., surge-longitudinal motion, sway-lateral motion, heave-vertical motion), and rotational (e.g., roll, pitch, yaw) degrees of freedom (DOF). A single-DOF translational motion system might provide a vibration sensation via of a "seat shaker." A common 6DOF configuration is a hexapod, which consists of a frame with six or more extendable struts (actuators) connecting a fixed base to a movable platform. In terms of actuators, electrical actuators are quiet and relatively maintenance free; however, they are not very responsive and they cannot hold the same load as can hydraulic or pneumatic systems. Hydraulic and pneumatic systems are smoother, stronger, and more accurate; however, they require compressors, which may be noisy. Servos are expensive and difficult to program.

Although other multimodal interactions are possible (e.g., gustatory, olfactory), there has been limited research and development beyond the primary three interaction modes (e.g., visual, auditory, haptic), although efforts have been made to support advances in olfactory interaction (Jones, Bowers, Washburn, Cortes, & Vijaya Satya, 2004).

Tracking systems. Tracking systems allow determination of users' head or limb position and orientation, or the location

of hand-held devices in order to allow interaction with virtual objects and traversal through 3D computer-generated worlds (Foxlin, 2002). Tracking is what allows the visual scene in a VE to coincide with a user's point of view, thereby providing an egocentric real-time perspective. Tracking systems must be carefully coupled with the visual scene, however, to avoid unacceptable lags (Kalawsky, 1993). Advances in tracking technology have been realized in terms of drift-corrected gyroscopic orientation trackers, outside-in optical tracking for motion capture, and laser scanners (Foxlin, 2002). The future of tracking technology is likely hybrid tracking systems, with an acoustic-inertial hybrid on the market (see http://www.isense.com/products/) and several others in research labs (e.g., magnetic-inertial, optical-inertial, and optical-magnetic). In addition, ultra-wideband radio technology holds promise for an improved method of omnidirectional point-to-point ranging.

Tracking technology also allows for gesture recognition, in which human position and movement are tracked and interpreted to recognize semantically meaningful gestures (Turk, 2002). Tracking devices that are worn (e.g., gloves, bodysuits) are currently more advanced than passive techniques (e.g., camera, sensors), yet the latter hold much promise for the future, as they are more powerful and less obtrusive than those that must be worn.

Interaction techniques. While one may think of joysticks and gloves when considering VE interaction devices, there are many techniques that can be used to support interaction with and traversal through a virtual environment. Interaction devices support traversal, pointing and selection of virtual objects, tool usage (e.g., through force and torque feedback), tactile interaction (e.g., through haptic devices), and environmental stimuli (e.g., temperature, humidity) (Bullinger, Breining, & Braun, 2001).

Supporting traversal throughout a VE, via motion interfaces, is of primary importance (Hollerbach, 2002). Motion interfaces are categorized as either active (e.g., locomotion) or passive (e.g., transportation). Active motion interfaces require self-propulsion to move about a virtual environment (e.g., treadmill, pedaling device, foot platforms). Passive motion interfaces transport users within a VE without significant user exertion (e.g., inertial motion, as in a flight simulator, or noninertial motion, such as in the use of a joystick or gloves). The utility, functionality, cost, and safety of locomotion interfaces beyond traditional options (e.g., joysticks) have yet to be proven. In addition, beyond physical training, concrete applications for active motion interfaces have yet to be clearly delineated.

Another interaction option is speech control (see chapters 8 and 35). Speaker-independent continuous speech recognition systems are currently commercially available (Huang, 1998). For these systems to provide effective interaction, however, additional advances are needed in acoustic and language-modeling algorithms to improve the accuracy, usability, and efficiency of spoken language understanding.

Gesture interaction allows users to interact through nonverbal commands conveyed via physical movement of the fingers, hands, arms, head, face, or other body limbs (Turk, 2002). Gestures can be used to specify and control objects of interest, direct navigation, manipulate the environment, and issue meaningful commands.

To support natural and intuitive interaction, a variety of interaction techniques can be coupled. Combining speech interaction with nonverbal gestures and motion interfaces can provide a means of interaction that closely captures real-world communications.

Augmented cognition techniques. Augmented cognition is a new computing paradigm in which users and computers are tightly coupled via physiological gauges that measure the cognitive state of users and adapt interaction to optimize human performance (Schmorrow, Stanney, Wilson, & Young, 2005). If incorporated into VE applications, augmented cognition could provide a means of evaluating their educational validity and compelling nature. Neuroscience studies have established that differential aspects of the brain are engaged when learning different types of materials and the areas in the brain that are activated change with increasing competence (Kennedy, Drexler, Jones, Compton, & Ordy, 2005). Thus, if VE users were immersed in an educational experience, augmented cognition technology could be used to gauge if targeted areas of the brain were being activated and dynamically modify the content of a VE learning curriculum if desired activation patterns were not being generated. Physiological measures could also be used to detect the onset of cybersickness (see "Cybersickness, Adaptation, and Aftereffects") and to assess the engagement, awareness, and anxiety of VE users, thereby potentially providing much more robust measures of immersion and presence (see "Sense of Pressence"). Such techniques could prove invaluable to entertainment VE applications (cf., Badiqué, Cavazza, Klinker, Mair, Sweeney, Thalmann, et al., 2002) that seek to provide the ultimate experience, military training VE applications (cf., Knerr, Breaux, Goldberg, & Thurman, 2002) that seek to emulate the "violence of action" found during combat, and therapeutic VE applications (cf., North, North, & Coble, 2002; Strickland, Hodges, North, & Weghorst, 1997) that seek to overcome disorders such as fear of heights or flying.

Software Requirements

Software development of VE systems has progressed tremendously, from proprietary and arcane systems, to development kits that run on multiple platforms (e.g., general-purpose operating systems to workstations) (Pountain, 1996). Virtual-environment system components are becoming modular and distributed, thereby allowing VE databases (e.g., editors used to design, build, and maintain virtual worlds) to run independently of visualizers and other multimodal interfaces via network links. Standard APIs (Application Program Interfaces) (e.g., OpenGL, Direct-3D, Mesa) allow multimodal components to be hardware-independent. Virtual environment programming languages are advancing, with APIs, libraries, and particularly scripting languages allowing nonprogrammers to develop virtual worlds (Stanney & Zyda, 2002). Advances are also being made in modeling of autonomous agents and communication networks used to support multiuser virtual environments.

Modeling. A VE consists of a set of geometry, the spatial relationships between the geometry and the user, and the

change in geometry invoked by user actions or the passage of time (Kessler, 2002). Generally, modeling starts with building the geometry components (e.g., graphical objects, sensors, viewpoints, animation sequences) (Kalawsky, 1993). These are often converted from CAD data. These components then get imported into the VE modeling environment and rendered when appropriate sensors are triggered. Color, surface textures, and behaviors are applied during rendering. Programmers control the events in a VE by writing task functions, which become associated with the imported components.

A number of 3D modeling languages and toolkits are available that provide intuitive interfaces and run on multiple platforms and renderers (e.g., 3D Studio Max, AC3D Modeler, ZBrush, modo 203, AccuRender, ACIS 3D, Ashlar-vellum Argon/Xenon/Cobalt, Carrara, CINEMA 4D, DX Studio, EON Studio, MultiGen Creator and Vega, RenderWare, solidThinking) (Ultimate 3D Links, 2007). In addition, there are scene management engines (e.g., R3vis Corporation's OpenRM Scene Graph is an API that provides cross-platform scene management and rendering services) that allow programmers to work at a higher level, defining characteristics and behaviors for more holistic concepts (Kershner, 2002; Menzies, 2002). There have also been advances in photorealistic rendering tools (e.g., Electric Image Animation System), which are evolving toward full-featured physics-based global illumination rendering systems (Heirich & Arvo, 1997; Merritt & Bacon, 1997). Taken together, these advances in software modeling allow for the generation of complex and realistic VEs that can run on a variety of platforms, permitting access to VE applications by both small- and large-scale application-development budgets.

Autonomous agents. Autonomous agents are synthetic or virtual human entities that possess some degree of autonomy, social ability, reactivity, and pro-activeness (Allbeck & Badler, 2002; also see chapter 15). They can have many forms (e.g., human, animal), which are rendered at various levels of detail and style, from cartoonish to physiologically accurate models. Such agents are a key component of many VE applications involving interaction with other entities, such as adversaries, instructors, or partners (Stanney & Zyda, 2002).

There has been significant research and development in modeling embodied autonomous agents. As with object geometry, agents are generally modeled off-line and then rendered during real-time interaction. While the required level of detail varies, modeling of hair and skin adds realism to an agent's appearance (Allbeck & Badler, 2002). There are a few toolkits available to support agent development, with one of the most notable offered by Boston Dynamics, Inc. (BDI) (http://www.bdi.com/), a spin-off from the MIT Artificial Intelligence Laboratory. BDI's products allow VE developers to work directly in a 3D database, interactively specifying agent behaviors, such as paths to traverse and sensor regions. The resulting agents move realistically, respond to simple commands, and travel about a VE as directed. Another option is ArchVision's (http://www.archvision.com/) 3D Rich Photorealistic Content (RPC) People.

Networks. Distributed networks allow multiple users at diverse locations to interact within the same virtual environment. Improvements in communications networks are required to allow realization of such shared experiences in which users, objects, processes, and autonomous agents from diverse locations interactively collaborate (Durlach & Mavor, 1995). Yet the foundation for such collaboration has been built within the Next Generation Internet (NGI). The NGI initiative (http://www.ngi.gov/) aimed to connect a number of universities and national labs at speeds 100 to 1,000 times faster than the 1996 Internet in order to experiment with collaborative-networking technologies, such as high-quality video conferencing and audio and video streams. The NGI's successor, the Large Scale Networking (LSN) Coordinating Group (http://www.hpcc.gov/iwg/lsn.html), aims to develop technologies and services that enable wireless, optical, mobile, and wireline communications and enhance networking software, with a goal of achieving the terabit per second objective which proved elusive to the NGI.

In addition, Internet2 is using existing networks (e.g., NSF's vBNS—Very high-speed Backbone Network Service) to determine the transport designs necessary to carry real-time multimedia data at high speed (http://apps.internet2.edu/). Distributed VE applications can leverage the special capabilities (e.g., high bandwidth, low latency, low jitter) of these advancing network technologies to provide shared virtual worlds (Singhal & Zyda, 1999).

DESIGN AND IMPLEMENTATION STRATEGIES

While many conventional HCI techniques can be used to design and implement VE systems, there are unique cognitive, content, product liability, and usage protocol considerations that must be addressed (see Fig. 31.2).

Cognitive Aspects

The fundamental objective of VE systems is to provide multimodal interaction or, when sensory modalities are missing, perceptual illusions that support human information processing in pursuit of a VE application's goals, which could range from training to entertainment. Ancillary yet fundamental to this goal is to minimize cognitive obstacles, such as navigational difficulties, that could render a VE application's goals inaccessible.

Multimodal interaction design. Virtual environments are designed to provide users with direct manipulative and intuitive interaction with multisensory stimulation (Bullinger et al., 2001). The number of sensory modalities stimulated and the quality of this multisensory interaction are critical to the realism and potential effectiveness of VE systems (Popescu, Burdea, & Trefftz, 2002). Yet there is currently a limited understanding of how to effectively provide such sensorial parallelism (Burdea, 1996); however, Stanney, Samman, Reeves, Hale, Buff, Bowers et al. (2004) provided a set of preliminary cross-modal integration rules. These rules consider aspects of multimodal interaction, including: (a) temporal and spatial coincidence, (b) working memory capacity, (c) intersensory facilitation effects, (d) congruency, and (e) inverse effectiveness. When multimodal sensory information is provided to users it is essential

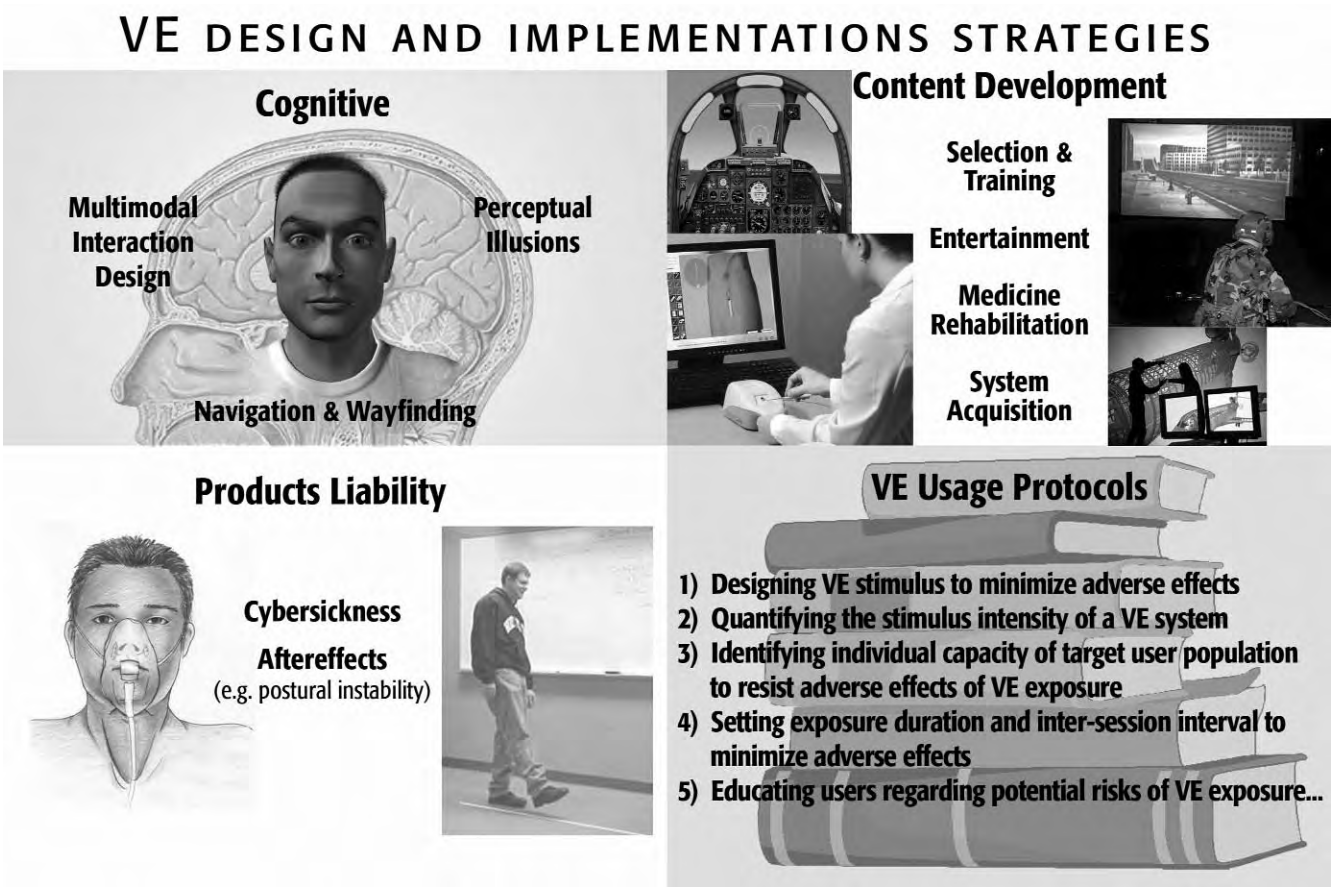


FIGURE 31.2. Design and implementation considerations for virtual environments.

to consider such rules governing the integration of multiple sources of sensory feedback. VE users have adapted their perception-action systems to "expect" a particular type of information flow in the real world; VEs run the risk of breaking these perception-action couplings if the full range of sensory is not supported or if it is supported in a manner that is not contiguous with real-world expectations. Such pitfalls can be avoided through consideration of the coordination between sensing and user command and the transposition of senses in the feedback loop. Specifically, command coordination considers user input as primarily monomodal and feedback to the user as multimodal. Designers need to consider which input modalities are most appropriate to support execution of a given task within the VE, if there is any need for redundant user input, and whether or not users can effectively handle such parallel input (Stanney, Mourant, & Kennedy, 1998; Stanney et al., 2004).

A limiting factor in supporting multimodal sensory stimulation in VEs is the current state of interface technologies. With the exception of the visual modality, current levels of technology simply cannot even begin to reproduce virtually those sensations, such as haptics and audition, which users expect in the real world. One solution to current technological shortcomings, sensorial transposition, occurs when a user receives feedback through senses other than those expected, which may occur because a command coordination scheme has substituted available sensory feedback for those that cannot be generated within a virtual environment. Sensorial substitution schemes may be one for one (e.g., visual for force) or more complex (e.g., visual for force and auditory; visual and auditory for force). If designed effectively, command coordination and sensory substitution schemes should provide multimodal interaction that allows for better user control of the virtual environment. On the other hand, if designed poorly, these solutions may in fact exacerbate interaction problems.

Perceptual illusions. When sensorial transpositions are used, there is an opportunity for perceptual illusions to occur. With perceptual illusions, certain perceptual qualities perceived by one sensory system are influenced by another sensory system (e.g., "feel" a squeeze when you see your hand "grabbing" a virtual object). Such illusions could simplify and reduce the cost of VE development efforts (Storms, 2002). For example, when attending to a visual image coupled with a low-quality auditory display, auditory-visual cross-modal perception allows for an increase in the perceived quality of the visual image. Thus, in this case if the visual image is the focus of the task, there may be no need to use a high-quality auditory display. Unfortunately, little is known about how to leverage these phenomena to reduce development costs while enhancing one's experience in a virtual environment. Perhaps the one exception is vection (e.g., a

compelling illusion of self-motion throughout a virtual world), which is known to be enhanced via a number of display factors, including a wide field of view and high spatial frequency content (Hettinger, 2002). Other such illusions exist (e.g., visual dominance) and could likewise be leveraged if perceptual and cognitive design principles are identified that can be used to trigger and capitalize on these illusory phenomena.

Navigation and wayfinding. Effective multimodal interaction design and use of perceptual illusions can be impeded if navigational complexities arise. Navigation is the aggregate of wayfinding (e.g., cognitive planning of one's route) and the physical movement that allows travel throughout a virtual environment (Darken & Peterson, 2002). A number of tools and techniques have been developed to aid wayfinding in virtual worlds, including maps, landmarks, trails, and direction finding. These tools can be used to display current position, current orientation (e.g., compass), log movements (e.g., "breadcrumb" trails), demonstrate or access the surround (e.g., maps, binoculars), or provide guided movement (e.g., signs, landmarks) (Chen & Stanney, 1999). Darken and Peterson (2002) provided a number of principles concerning how best to use these tools. If effectively applied to VEs, these principles should lead to reduced disorientation and enhanced wayfinding in large-scale virtual environments.

Content Development

Content development is concerned with the design and construction of the virtual objects and synthetic environment that support a VE experience (Isdale, Fencott, Heim, & Daly, 2002). While this medium can leverage existing HCI design principles, it has unique design challenges that arise due to the demands of real-time, multimodal, collaborative interaction. In fact, content designers are just starting to appreciate and determine what it means to create a full sensory experience with user control of both point of view and narrative development. Aesthetics is thought to be a product of agency (e.g., pleasure of being), narrative potential, presence and co-presence (e.g., existing in and sharing the virtual experience), as well as transformation (e.g., assuming another persona) (Church, 1999; Murray, 1997). Content development should be about stimulating perceptions (e.g., sureties, surprises), as well as contemplation over the nature of being (Fencott, 1999; Isdale et al., 2002).

Existing design techniques, for example from entertainment, video games, and theme parks can be used to support VE content development (see chapters 43 and 46). Game development techniques that can be leveraged in VE content development include but are not limited to providing a clear sense of purpose, emotional objectives, perceptual realism, intuitive interfaces, multiple solution paths, challenges, a balance of anxiety and reward, as well as an almost unconscious flow of interaction (Isdale et al., 2002). From theme park design, content development suggestions include: (a) having a story that provides the all-encompassing theme of the VE and thus the "rules" that guide design, (b) providing location and purpose, (c) using cause-and-effect to lead users to their own conclusions, and (d) anchoring users in the familiar (Carson, 2000a, 2000b). While these suggestions provide guidelines for VE content development, considerable creativity is still an essential component of the process. Isdale et al. (2002) suggested that the challenges of VE content development highlight the need for art to compliment technology.

Products Liability

Those who implement VE systems must be cognizant of potential product-liability concerns. Exposure to a VE system often produces unwanted side effects that could render users incapable of functioning effectively upon return to the real world. These adverse effects may include nausea and vomiting, postural instability, visual disturbances, and profound drowsiness (Stanney, Salvendy, Deisigner, DiZio, Ellis, Ellison, et al., 1998). As users subsequently take on their normal routines, unaware of these lingering effects, their safety and well-being may be compromised. If a VE product occasions such problems, liability of VE developers or system administrators could range from simple accountability (e.g., reporting what happened) to full legal liability (e.g., paying compensation for damages) (Kennedy & Stanney, 1996; Kennedy, Kennedy, & Bartlett, 2002). In order to minimize their liability, manufacturers and corporate users should design systems and provide usage protocols to minimize risks, warn users about potential aftereffects, monitor users during exposure, assess users' risk, and debrief users after exposure.

Usage Protocols

In order to minimize products liability concerns, VE usage protocols should be carefully designed. A comprehensive VE usage protocol will involve the following activities (see Stanney, Graeber, & Kennedy, in press):

1. Designing VE stimulus to minimize adverse effects:
   - Ensure system lags/latencies are minimized and stable;
   - Ensure frame rates are optimized;
   - Ensure interpupilary distance of visual display is adjustable and set appropriately;
   - Determine if size of field of view is causing excessive vection, such that the spatial frequency content of the visual scene should be reduced;
   - Determine if multimodal feedback is integrated into the VE such that sensory conflicts are minimized.
2. Quantifying stimulus intensity of a VE system using the Simulator Sickness Questionnaire (Kennedy, Lane, Berbaum, & Lilienthal, 1993) or other means and comparing the outcome to other systems (see Stanney et al., 2005). If a given VE system is of high intensity (say the 50th or higher percentile) and is not redesigned to lessen its impact, significant dropouts can be expected.
3. Identifying individual capacity of target user population to resist adverse effects of VE exposure via the Motion History Questionnaire (Kennedy & Graybiel, 1965) or other means.

- For highly susceptible population, redesigning the VE to reduce stimulus intensity or expect high dropout rates;
- Providing warnings for those with severe susceptibility to motion sickness, seizures, migraines, cold, flu, or other ailments.

4. Setting exposure duration and intersession interval to minimize adverse effects by (see Stanney et al., 2005):

   - Limiting initial exposures (e.g., 10 minutes or less);
   - Setting intersession exposure intervals two to five days apart to enhance individual adaptability;
   - Incrementally increasing VE stimulus intensity within one exposure (incremental adaptation) or across multiple exposures (incremental habituation);
   - Determining if users are able to complete an adaptive process at each increment of stimulus intensity (e.g., depressing sensitization to preexposure levels or below); if not, lowering stimulus intensity;
   - Avoiding repeated exposure intervals occurring less than two hours apart if adverse effects are experienced in an exposure.

5. Educating users regarding potential risks of VE exposure (e.g., inform users they may experience nausea, malaise, disorientation, headache, dizziness, vertigo, eyestrain, drowsiness, fatigue, pallor, sweating, increased salivation, and vomiting).
6. Educating users regarding potential adverse aftereffects of VE exposure (e.g., inform users they may experience disturbed visual functioning, visual flashbacks, and unstable locomotor and postural control for prolonged periods postexposure).
7. Instructing users to terminate VE interaction if they start to feel ill.
8. Providing adequate air flow and comfortable thermal conditions.
9. Adjusting equipment to minimize fatigue.
10. For strong VE stimuli, warning users to avoid extraordinary maneuvers (e.g., flying backward or experiencing high rates of linear or rotational acceleration) during initial interaction.
11. Providing an attendant to monitor users' behavior and ensure their well-being:

    - Attendant should terminate exposure if they see red flags (e.g., excessive sweating, verbal frustration, prolonged lack of movement in VE, and less overall movement [e.g., restricting head movement] or if users verbally request termination or complain of excessive symptoms);
    - Attendant should assess well being of users' postexposure (e.g., can use battery similar to field sobriety tests; Tharp, Burns, & Moskowitz, 1981).

12. Specifying amount of time postexposure that users must remain on premises before driving or participating in other such high-risk activities. Do not allow individuals who fail postexposure tests or experience adverse aftereffects to conduct high-risk activities until they have recovered (e.g., have someone else drive them home).
13. Calling users the next day or having them call to report any prolonged adverse effects.

Regardless of the strength of the stimulus or the susceptibility of the user, following a systematic usage protocol can minimize the adverse effects associated with VE exposure.

HEALTH AND SAFETY ISSUES

The health and safety risks associated with VE exposure complicate usage protocols and lead to products liability concerns. It is thus essential to understand these issues when utilizing VE technology. There are both physiological and psychological risks associated with VE exposure, the former being related primarily to sickness and aftereffects and the latter primarily being concerned with the social impact.

Cybersickness, Adaptation, and Aftereffects

Motion-sickness-like symptoms and other aftereffects (e.g., balance disturbances, visual stress, altered hand-eye coordination) are unwanted byproducts of VE exposure. The sickness related to VE systems is commonly referred to as "cybersickness" (McCauley & Sharkey, 1992). Some of the most common symptoms exhibited include dizziness, drowsiness, headache, nausea, fatigue, and general malaise (Kennedy et al., 1993). More than 80% of users will experience some level of disturbance, with approximately 12% ceasing exposure prematurely due to this adversity (Stanney, Kingdon, Nahmens, & Kennedy, 2003). Of those who drop out, approximately 10% can be expected to have an emetic response (e.g., vomit), however, only 1–2% of all users will have such a response. These adverse effects are known to increase in incidence and intensity with prolonged exposure duration (Kennedy, Stanney, & Dunlap, 2000). While most users will experience some level of adverse effects, symptoms vary substantially from one individual to another, as well as from one system to another (Kennedy & Fowlkes, 1992). These effects can be assess via the Simulator Sickness Questionnaire (Kennedy et al., 1993), with values above 20 requiring due caution (e.g., warn and observe users) (Stanney et al., 2005).

To overcome such adverse effects, individuals generally undergo physiological adaptation during VE exposure. This adaptation is the natural and automatic response to an intersensorily imperfect VE and is elicited due to the plasticity of the human nervous system (Welch, 1978). Due to technological flaws (e.g., slow update rate, sluggish trackers), users of VE systems may be confronted with one or more intersensory discordances (e.g., visual lag, a disparity between seen and felt limb position). In order to perform effectively in the VE, they must compensate for these discordances by adapting their psychomotor behavior or visual functioning. Once interaction with a VE is discontinued, these compensations persist for some time after exposure, leading to aftereffects.

Once VE exposure ceases and users return to their natural environment they are likely unaware that interaction with the VE has potentially changed their ability to effectively interact with their normal physical environment (Stanney & Kennedy, 1998). Several different kinds of aftereffects may persist for prolonged periods following VE exposure (Welch, 1997). For

example, hand-eye coordination can be degraded via perceptual-motor disturbances (Kennedy, Stanney, Ordy, & Dunlap, 1997; Rolland, Biocca, Barlow, & Kancherla, 1995), postural sway can arise (Kennedy & Stanney, 1996), as can changes in the vestibulo-ocular reflex (VOR), or one's ability to stabilize an image on the retina (Draper, Prothero, & Viirre, 1997). The implications of these aftereffects are that:

1. VE exposure duration may need to be minimized;
2. Highly susceptible individuals or those from clinical populations (e.g., those prone to seizures) may need to avoid or be banned from exposure;
3. Users should be closely monitored during VE exposure;
4. Users' activities should be closely monitored for a considerable period of time postexposure to avoid personal injury or harm.

Social Impact

Virtual environment technology, like its ancestors (e.g., television, computers), has the potential for negative social implications through misuse and abuse (Kallman, 1993; also see chapter 61). Yet violence in VE is nearly inevitable, as evidenced by the violent content of popular video games. Such animated violence is a known favorite over the portrayal of more benign emotions such as cooperation, friendship, or love (Sheridan, 1993; also see chapter 4). The concern is that users who engage in what seems like harmless violence in the virtual world may become desensitized to violence and mimic this behavior in the look-alike real world.

Currently, it is not clear whether or not such violent behavior will result from VE exposure, early research, however, is not reassuring. Calvert and Tan (1994) found VE exposure to significantly increase the physiological arousal and aggressive thoughts of young adults (Calvert & Tan, 1994). Perhaps more disconcerting was that neither aggressive thoughts nor hostile feelings were found to decrease due to VE exposure, thus providing no support for catharsis. Such increased negative stimulation may then subsequently be channeled into real-world activities. The ultimate concern is that VE immersion may potentially be a more powerful perceptual experience than past, less interactive technologies, thereby increasing the negative social impact of this technology (Calvert, 2002). A proactive approach is needed which weighs the risks and potential consequences associated with VE exposure against the benefits. Waiting for the onset of harmful social consequences should not be tolerated.

VIRTUAL ENVIRONMENT USABILITY ENGINEERING

Most VE user interfaces are fundamentally different from traditional graphical user interfaces, with unique I/O devices, perspectives, and physiological interactions. Thus, when developers and usability practitioners attempt to apply traditional usability engineering methods to the evaluation of VE systems they find few if any that are particularly well suited to these environments (for notable exceptions see Gabbard, Hix, & Swan, 1999; Hix & Gabbard, 2002; Stanney, Mollaghasemi, & Reeves, 2000). There is a need to modify and optimize available techniques to meet the needs of VE usability engineering, as well as to better characterize factors unique to VE usability, including sense of presence and VE ergonomics.

Usability Techniques

Assessment of usability for VE systems must go beyond traditional approaches, which are concerned with the determination of effectiveness, efficiency, and user satisfaction (see chapter 55). Evaluators must consider whether multimodal input and output is optimally presented and integrated, navigation is supported to allow the VE to be readily traversed, object manipulation is intuitive and simple, content is immersive and engaging, and the system design optimizes comfort while minimizing sickness and aftereffects. The affective elements of interaction become important when evaluating VE systems (see chapter 58). It is an impressive task to ensure that all of these criteria are met.

Gabbard, Hix, and Swan (1999) have developed a taxonomy of VE usability characteristics that can serve as a foundation for identifying and evaluating usability criteria particularly relevant to VE systems. Stanney, Mollaghasemi, and Reeves (2000) used this taxonomy as the foundation on which to develop an automated system, *MAUVE* (Multicriteria Assessment of Usability for Virtual Environments), which assesses VE usability in terms of how effectively each of the following are designed: (a) navigation, (b) user movement, (c) object selection and manipulation, (d) visual output, (e) auditory output, (f) haptic output, (g) presence, (h) immersion, (i) comfort, (j) sickness, and (k) aftereffects. *MAUVE* can be used to support expert evaluations of VE systems, similar to the manner in which traditional heuristic evaluations are conducted. Due to such issues as cybersickness and aftereffects, it is essential to use these or other techniques (cf., Modified Concept Book Usability Evaluation Methodology; Swartz, 2003) to ensure the usability of VE systems, not only to avoid rendering them ineffective but also to ensure that they are not hazardous to users.

Sense of Presence

One of the usability criteria unique to VE systems is sense of presence. Virtual environments have the unique advantage of leveraging the imaginative ability of individuals to psychologically "transport" themselves to another place, one that may not exit in reality (Sadowski & Stanney, 2002). To support such transportation, VEs provide physical separation from the real world, by immersing users in the virtual world via an HMD, then imparting sensorial sensations via multimodal feedback that would naturally be present in the alternate environment. Focus on generating such presence is one of the primary characteristics distinguishing VEs from other means of displaying information.

Presence has been defined as the subjective perception of being immersed in and surrounded by a virtual world rather

than the physical world one is currently situated in (Stanney, Salvendy, et al., 1998). Virtual environments that engender a high degree of presence are thought to be more enjoyable, effective, and well received by users (Sadowski & Stanney, 2002). To enhance presence, designers of VE systems should spread detail around a scene, let user interaction determine when to reveal important aspects, maintain a natural and realistic, yet simple appearance, and utilize textures, colors, shapes, sounds, and other features to enhance realism (Kaur, 1999). To generate the feeling of immersion within the environment, designers should isolate users from the physical environment (use of an HMD may be sufficient), provide content that involves users in an enticing situation supported by an encompassing stimulus stream, provide natural modes of interaction and movement control, and utilize design features that enhance vection (Stanney et al., 2000). Presence can be assessed via Witmer and Singer's (1998) Presence Questionnaire or techniques used by Slater and Steed (2000), as well as a number of other means (Sadowski & Stanney, 2002).

Virtual Environment Ergonomics

Ergonomics, which focuses on fitting a product or system to the anthropometric, musculoskeletal, cardiovascular, and psychomotor properties of users, is an essential element of VE system design (McCauley-Bell, 2002). Supporting user comfort while donning cumbersome HMDs or unwieldy peripheral devices is an ergonomics challenge of paramount importance; discomfort could supersede any other sensations (e.g., presence, immersion). If a VE produces discomfort, participants may limit their exposure time or possibly avoid repeat exposure. Overall physical discomfort should thus be minimized, while user safety is maximized.

Ergonomic concerns affecting comfort include visual discomfort resulting from visual displays with improper depth cues, poor contrast and illumination, or improperly set IPDs (Stanney et al., 2000). Physical discomfort can be driven by restrictive tethers; awkward interaction devices; or heavy, awkward, and constraining visual displays. To enhance the ergonomics of VE systems, several factors should be considered, including (McCauley-Bell, 2002):

- Is operator movement inhibited by the location, weight, or window of reach of interaction devices or HMDs?
- Does layout and arrangement of interaction devices and HMDs support efficient and comfortable movement?
- Is any limb overburdened by heavy interaction devices or HMDs?
- Do interaction devices require awkward and prolonged postures?
- If a seat is provided, does it support user movement and is it of the right height with adequate back support?
- If active motion interfaces are provided (e.g., treadmills), are they adjustable to ensure fit to the anthropometrics of users?
- Are the noise and sound levels within ergonomic guidelines and do they support user immersion?

APPLICATION DOMAINS

The wide range of VE types and designs makes them ideal tools for supporting a range of performance-enhancing tasks. Most commonly, VEs are used as training aids; a less common but increasingly well-supported use is as a selection tool, the flip side to the training application. Other applications for VE include their growing potential for providing entertainment, their use as medical rehabilitative tools, and even their implementation within the training system acquisition cycle of the U.S. government.

VE as a Selection and Training Tool

Perhaps the most common application for VEs is as training tools. The U.S. military has focused considerable resources on VE as a means to resolve many of the training deficits that result from the rigors of military life (e.g., sustaining schoolhouse-gleaned skills and knowledge sets during prolonged deployment periods, acquiring new skills and knowledge while away from the schoolhouse, providing large-scale environments for multitrainee exercises, distributing training exercises). As VE training exercises become more closely aligned to real-world conditions, a series of factors, including training transfer effectiveness, cost, logistics and safety, move to the forefront. Further, in terms of applying VE to enhance human performance, training is actually the second stage of a two-stage process. Ideally, one would like to ensure that those individuals for whom training is being provided have a certain degree of "performance capability." In this approach, VE systems can be used as part of a comprehensive performance-enhancement program that focuses on selecting those users with the correct set of Knowledge, Skills, and Abilities (KSAs) and then providing, when needed, training to fine-tune those KSAs.

Given the potentially high cost of developing VEs, this makes sense. Given the broader, yet equally high costs of bringing individuals into an organization and training them to some level of proficiency, it becomes glaringly obvious that having an ability to select individuals who are most likely to benefit from VE training is critical. For example, in the U.S. Air Force the cost of a single individual student pilot failing to complete basic flight school can run to $100,000 (Siem, Carretta, & Mercatante, 1988). Student failure can be attributed to both inadequate selection techniques and deficient training techniques. Clearly, both selection and training play critical roles in producing effective users. The challenge is to develop a program that ensures a smooth union between the two-one, which identifies the best candidates and then provides the optimum training.

Traditional approaches to selection focus on social and psychophysical assessments. For example, aptitude tests, ranging from traditional pen-and-paper-type efforts (Carretta & Ree, 1995) to psychomotor tests (Carretta & Ree, 1993) to computer-based (but not VE!) assessments (Carretta & Ree, 1993) have all been used with varying levels of success. The single most important criticism of each of these approaches is that they are designed to be predictive of future performance, and as such

are more often than not abstractions of aspects of the larger task(s) for which the individual is being selected. An alternate approach would be to provide selectees with a method that provides a direct indication of their performance abilities. This distinction, essentially between a test being predictive of performance ability versus indicative of performance ability, has a great impact on selection. A meta-analysis performed within the aviation domain, where much of selection research has focused, found that typical predictive validities (most often reported as either the correlation coefficient, *r*, or the multiple correlation coefficient, *R*, and representing the degree to which a given predictor/set of predictors and performance metrics are related) for such assessments range from a low of 0.14 to a "high" of about 0.40 (Martinussen, 1996). Yet, when a simulation component is added to this mix, these values have been shown to improve considerably, pushing correlations towards the 0.60 level (Gress & Willkomm, 1996).

A potential deterrent with using VEs as part of a selection toolkit is the high cost associated with developing both the system as well as the performance measures that need to be integrated with it. However, when considered as one part of a comprehensive performance enhancement program, this concern vanishes. The same effort and cost devoted to developing the system for selection can typically, with minimal additional input, be modified (e.g., via altered scenarios) to meet identified training goals and objectives. It is therefore necessary to explore some of the basic requirements for using VEs as training tools in order to understand how one technology can truly be applied for these multiple purposes.

A constant thread in training research is the notion that in order for training to be effective, the basic skills being taught must show some degree of transfer to real-world performance. Over 100 years ago, Thorndike and Woodworth (1901) laid down the most basic training-transfer principle when they proposed that transfer was determined by the degree of similarity between any two tasks. Applying this heuristic to VE design, one might conclude that the most basic way to ensure perfect transfer is to ensure that the real-world performance elements that are meant to be trained should be replicated perfectly in a virtual environment. This notion of "identical elements" could easily create a serious challenge for system designers even by today's technology standards, as VEs are still not able to perfectly duplicate the wide range of sensorial stimuli encountered during daily interactions with our world (Stoffregen, Bardy, Smart, & Pagulyan, 2003). Countering this somewhat simplistic design approach is Osgood's (1949) principle that greater similarity between any two tasks along certain dimensions will not guarantee wholesale, perfect transfer. The challenge, as noted by Roscoe (1982) is to find the right balance between technical fidelity and training effectiveness.

This suggests that when developing training systems, it is important to take two factors into consideration. The first is how one can measure training transfer, and the second is how one can assess the cost/benefit relationship associated with making the decision to use VEs to provide training. The most basic metric for assessing transfer is the percent of transfer (Wickens & Hollands, 2000) in which the amount of time saved in learning the real world task is expressed in terms of how much time a control group, denied such training, required to learn the task (see Equation 1):

$$\%Transfer = \{(Control\ Time - Transfer\ Time)/ Control\ Time\}*100 \quad (1)$$

A quick inspection of this metric suggests that it is overly simplistic. Consider a situation in which the percent transfer is high, yet the total amount of time spent training in the transfer platform (e.g., VE) may be greater than that spent by a control group to achieve similar levels of performance. Such an outcome is hardly cost effective. Consequently, in parallel to a basic quantification of transfer, another metric, the Training Effectiveness Ratio (TER; Povenmire & Roscoe, 1973) focusing on training efficiency is needed (see Equation 2).

$$TER = Amount\ of\ time\ saved\ learning\ task\ by\ Transfer\ Group/Time\ spent\ in\ transfer\ platform \quad (2)$$

All things being equal, these two measures should perfectly quantify VE utility as a training tool. Yet, a third factor, cost, almost always factors in. Indeed, cost is often the single most critical driving factor in determining whether to supplement a training program with VE tools. In terms of training effectiveness, cost can be quantified using the Training Cost Ratio (TCR; Wickens & Hollands, 2000; see Equation 3).

$$TCR = Training\ cost\ in\ real\ platform/ Training\ cost\ in\ transfer\ platform \quad (3)$$

When making the determination to use a VE to provide some (or, less likely, all) of a given set of training, one must consider the interplay between these measures. Specifically, the relationship between training effectiveness, as expressed through the TER, and training cost, as expressed through the TCR, will drive this decision. A good rule of thumb, provided by Wickens and Holland (2000) is:

$$TER\ X\ TCR > 1.0,\ VE\ Training\ is\ effective \quad (4)$$

$$TER\ X\ TCR < 1.0,\ VE\ Training\ is\ ineffective \quad (5)$$

A slightly different way of interpreting this was provided by Roscoe (1982) who suggested that, given that the higher the level of technical fidelity the more costly the system, there is a region of design space within which cost/benefit is optimal; move away from this region in either direction, and you run the risk of paying a lot for a system that delivers less than effective (and perhaps even negative) training.

This brief overview should not leave one with the impression that the decision to use—or to not use—VEs for training is simple. Assessing the TER * TCR function is no easy matter. Simply put, the cost of performing the study needed to capture the effectiveness of a training device can far outstrip the perceived immediate impact of incorporating the device into a training curriculum. Often, the type of system being developed requires a significant investment in time, money, and labor in order to be evaluated. Consider a situation in which a new flight simulator is

being considered for purchase. In order to determine both the TER and the TCR, one must conduct an actual transfer study, which involves

1. Removing a set of trainees from their curriculum for experimental purposes;
2. Developing reliable performance metrics;
3. Providing both the experiment group and the control group with adequate time in the real world (e.g., aircraft) environment and the experiment group time in the VE;
4. Dealing with typically low effect sizes, which forces experimenters to use large sample sizes.

Yet, just as the TER * TCR function could guide when to use and not use VEs, it is possible to pull some general principles from the literature. For example, cognitively oriented training, which often features problem solving or decision making as a key training goal, may oftentimes be satisfied by using simple, relatively inexpensive visualization systems (Gopher, Weil, & Bareket, 1994; Morris & Tarr, 2002; Stone, 2002; Figueroa, Bischof, Boulanger, & Hoover, 2005; Milham, Hale, Stanney, Cohn, Darken, & Sullivan, 2004). Contrastingly, motor skill based training, which often requires complex interactions between users and their environments, frequently requires more costly solutions. When considering whether or not to "go virtual," a useful solution would be to first determine the general category of skill to be trained (Cognitive or Motor; cf., Anderson, 1987); estimate the TER and TCR from sources as broadly ranging as basic science studies, published costs of similarly desired systems, and one's own best estimates; and then develop a basic decision matrix to establish a range of TER * TCR values.

VE as Entertainment Tool

In many ways entertainment can be considered a prime technology pusher. Consider the evolution of the VCR, Digital Video Disk, High Definition Television and, more recently, the advent of the all-in-one device (phone, PDA, music/video player) in various forms. Each of these devices captured a unique market niche precisely because they pushed ahead current technology barriers in ways never before considered. Yet, if entertainment provides the technology push necessary to achieve revolutions, the consumers of such technologies provide an equally strong technology pull, placing discrete constraints on this transformation. Often, such constraints are directly at odds with entertainment's evolution. An entertainment developer may create a new game that requires a high-end graphics system, a costly sound system, and perhaps even a tactile (haptic) interface to fully take advantage of the system. The consumers, on the other hand, may have their own satisficing heuristic, requiring that, in addition to providing novel entertainment value, any new technology must cost less than a fixed amount and fit into their back pockets. Unless there is significant demand for a technology to factor these constraints into its development (e.g., a significant consumer base) the resultant product may either be forestalled or, when placed on the market, flounder and vanish overnight.

Until recently, the high cost of the component technologies comprising VEs precluded their becoming anything more than a research test-bed for well-funded labs or an exclusive tool for high-end consumers, like commercial airline companies or the military establishments of entire countries. Yet, after nearly four decades of research and development, VEs have finally begun to realize their potential. This maturation owes much to advancements in the computer industry which, proceeding at a Moore's Law rate, has been able to evolve the simple desktop PC into a powerful graphical rendering system. As technology barriers have been shattered, so too have many of the HCI challenges that were typically associated with earlier technology-limited systems, thus resulting in a number of entertainment-based applications. Current VE entertainment applications include games, online communities, location-based entertainment, theme parks, and other venues (Badiqué et al., 2002). From interactive arcades to cybercafes, the entertainment industry has leveraged the unique characteristics of the VE medium, providing dynamic and exciting experiences. Such applications tend to set the pace, both technological and in terms of content, for advances in VE technology in general. Loosely put, the goal of such entertainment technologies is to provide consumers with experiences that they would not ordinarily encounter in their typical daily activities. Consequently, the goals and objectives of entertainment-based systems are quite broadly defined (and, sometimes, not necessarily defined at all!) and shaped primarily through user feedback vis-à-vis factors such as "ease of use," "engagement level," "interest level," and so forth. These factors are molded to a large extent by popular culture. By attending to these global trends, developers can determine what technologies exist, or need to be created, to support current user demand. Thus, in spite of the optimism generated by such entertainment applications, a cautionary note is warranted. The uses for which typical VEs have been, and continue to be, considered (e.g., primarily as training tools) have a strong scientific basis upon which to define system specifications, and a lengthy history linking these specifications to specific technologies. Consider, for example, the prerequisites for designing a system to train a Naval Officer to maneuver a ship in a harbor (Hays, Castillo, Seamon, & Bradley, 1997; Nguyen, Cohn, Mead, Helmick, & Patrey, 2001). Since the trainees will, ultimately, be graded on their performances, developers must identify the necessary training goals and objectives (e.g., Osgood's [1949] identical elements), identify the necessary technologies (or develop them if absent), integrate them, test and validate their training efficacy, and ultimately hand them off to be used. Such a developmental process can require tens of millions of dollars, time investments on the order of years, and personnel investments numbering in the dozens. Nevertheless, precisely because training is perceived as (and often is) critical to a given organization's success, training requirements can provide a justification for such large investments. On the other hand, in entertainment, the driving force is delivering customer satisfaction ahead of the competition. As a result, it is quite likely that the vast majority of entertainment developers will assume less risk when developing their products in order to cater to a constantly shifting market interest.

VE as a Medical Tool

Much has been written about applications for VE within the medical arena (cf., Moline, 1995; Satava & Jones, 2002). While some of these applications represent unique approaches to harnessing the power of VE, many other applications, such as simulating actual medical procedures, reflect training applications, and therefore will not be discussed anew here. One area of medical application for which VE is truly coming into its own is medical rehabilitation. In particular, two areas of rehabilitation, psychological/cognitive and motor, show strong promise.

Psychological/cognitive rehabilitation applications. Perhaps the fastest-growing application for VEs in medicine is in the area of phobia treatment (Rizzo, Buckwalter, & van der Zaag, 2002; Satava & Jones, 2002; Thacker, 2003). The reason for this is likely due to the ideal matching between VE's strengths (presenting evolving information with which users can interact in various ways) and such therapy's basic requirements (incremental exposure to the offending environment). Importantly, compared to previous treatment regimens, which oftentimes simply required patients to mentally revisit their fears, VEs offer a significantly more immersive experience. In fact, it is quite likely that many of VE's shortcomings, such as poor visual resolution, inadequate physics modeling underlying environmental cues, and failure to fully capture the wide range of sensorial cues present in the real world will be ignored by the patient, whose primary focus is on overcoming anxiety engendered by her or his specific phobias. On a practical level, VEs enable patients to simply visit their therapist's office, where they can be provided an individually tailored multimodal treatment experience (Rothbaum, Hodges, Watson, Kessler, & Opdyke, 1996; Emmelkamp, Bruynzeel, & Drost, van der Mast, 2001; Anderson, Rothbaum, & Hodges, 2003).

Motor rehabilitation applications. Many of VE's qualities that make it an ideal tool for providing medical training—such as tactile feedback and detailed visual information (Satava & Jones, 2002)—also make it an ideal candidate for supplementing motor rehabilitation treatment regiments for such conditions as stroke, Parkinson's, and pain treatment (Holden, Todorov, Callahan, & Bizzi, 1999; Gershon, Zimand, Lemos, Rothbaum, & Hodges, 2003). In determining how best to apply VE in these and other treatment regimens, Holden (2005) suggested considering three practical areas in which VE is strongest: repetition, feedback, and motivation. All three elements are critical to both effective learning and regaining of motor function; the application of VE, in each case, provides a powerful method for rehabilitation specialists to maximize the effect of a treatment regimen for a given session, and, because they may reduce the time investment required by therapists (one can simply immerse the patient, initiate the treatment, and then allow the program to execute), to also expand the access of such treatments to a wider population.

Since VE is essentially computer-based, patients can effectively have their attention drawn to a specific set of movement patterns they may need to make to regain function; conducting this in a "loop" provides unlimited ability to repeat a pattern, while using additional visualization aids, such as a rendered cursor or "follow-me" types of cues, to force the patient into moving a particular way (cf., Chua, Crivella, Daly, Hu, Schaaf, Ventura, et al., 2003). As well, it is a relatively simple matter to digitize movement information, store it, and then, based on comparisons to previously stored, desired, movement patterns provide additional feedback to assist the patient. In terms of motivation, treatment scenarios can be tailored to capture specific events that individual patients find most motivating: a baseball fan can practice her movement in a baseball-like scenario; a car enthusiast can do so in a driving environment.

There are certain caveats that must be considered when exploiting VE for rehabilitation purposes, most significantly the potentially rapid loss of motor adaptations following VE exposure. Lackner and DiZio (1994) demonstrated that certain basic patterns of sensori-motor recalibrations learned in a given physical environment can diminish within an hour, postexposure, although subsequent findings (DiZio & Lackner, 1995) suggest that there are certain transfer benefits that are longer-lasting. Brashers-Krugg, Shadmehr, and Bizzi (1996) provided additional evidence that sensori-motor recalibrations of the type likely to be required for rehabilitation have post exposure periods, in excess of four hours, during which their effects can be extinguished. Most importantly, Cohn, DiZio and Lackner (2000) demonstrated that such recalibrations, when learned in VE, have essentially no transfer to real-world conditions postexposure. Clearly, more research is needed to understand the conditions under which such transfer effects can be made most effective within the clinical setting.

VE as a System Acquisition Tool

In many areas of research and development, rising costs, with their associated increasing consequences of failure, have all but forced scientists and engineers to focus only on developing those products that have an almost guaranteed chance of succeeding in the marketplace. While this may seem like an ideal situation—after all, who would want to field technologies that won't work? —the net result is an increasing aversion, among companies and their financial backers, to shoulder much risk. Consequently, much-needed breakthrough technologies may be ignored in favor of "sure-thing" technologies, which more often than not do little to push ahead technology barriers.

The United States Department of Defense (DoD), seeking to reverse this trend while maintaining a level of control over risk, developed the process of Simulation Based Acquisition (SBA), a methodology that introduced the use of modeling and simulation (M&S) within the product-development process (Sanders, 1997; Zittel, 2001). Most recently, these M&S solutions have become increasingly integrated into 4D interactive tools that enable human-in-the-loop testing of design and operational principles (Sanders, 1997). As one example that cuts across DoD and industry, consider the development of their newest fighter aircraft, the Joint Strike Fighter (JSF). While more common M&S solutions, such as the ability to simulate individual components digitally before manufacturing them (through the CAD/CAM process), formed a core component of the development process, more advanced M&S solutions, such as VEs, have enabled developers to include users throughout the develop-

ment process, in order to answer such questions as (a) how users (e.g., pilots) would implement the fighter in combat, using virtual war-gaming techniques; (b) how well individual pilots could handle a given JSF design; and, (c) how the JSF would alter the nature of warfare (Zittel, 2001).

CONCLUSIONS

Clearly, VEs have evolved significantly from Morton Heilig's 1962 Sensorama experience (Burdea & Coiffet, 2003). Yet, despite significant revolutions in component technology, many of the challenges addressed by that primitive system, such as multimodal sensori-interaction, visual representation, and scenario generation, have yet to be fully resolved more than 40 years later. At the same time, our understanding of the potential such tools have to offer has advanced considerably. No longer simple amusements, these powerful machines can provide educational value, assist in treating core physical and cognitive maladies, and even help design better VE systems. As the uses for which VEs are ideally suited continue to be defined and refined, one can anticipate that current development challenges will be resolved, only to find new ones waiting to take their place.

ACKNOWLEDGMENTS

The author would like to thank Branka Wedell for her contribution of Figs. 1 and 2. This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. IRI-9624968, the Office of Naval Research (ONR) under Grant No. N00014-98-1-0642, the Naval Air Warfare Center Training Systems Division (NAWC TSD) under contract No. N61339-99-C-0098, and the National Aeronautics and Space Administration (NASA) under Grant No. NAS9-19453. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views or the endorsement of the NSF, ONR, NAWC TSD, or NASA.

References

Akamatsu, M. (1994, July). Touch with a mouse. A mouse type interface device with tactile and force display. *Proceedings of 3rd IEEE International Workshop on Robot and Human Communication,* Nagoya, Japan, 140–144.

Algazi, V. R., Duda, R. O., Thompson, D. M., & Avendano, C. (2001, October). The CIPIC HRTF Database. *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Electroacoustics,* New Paltz, NY, 99–102.

Allbeck, J. M., & Badler, N. I. (2002).Embodied autonomous agents. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 313–332). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 94*, 192–210.

Anderson, P., Rothbaum, B. O., & Hodges, L. F. (2003). Virtual reality exposure in the treatment of social anxiety. *Cognitive and Behavioral Practice, 10*(3), 240–247.

Badiqué, E., Cavazza, M., Klinker, G., Mair, G., Sweeney, T., Thalmann, D., & Thalmann, N. M. (2002). Entertainment applications of virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 1143–1166). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Begault, D. (1994). *3-D Sound for virtual reality and multimedia*. Boston: Academic Press.

Biggs, S. J., & Srinivasan, M. A. (2002). Haptic interfaces. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 93–116). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Brashers-Krug, T., Shadmehr, R., & Bizzi, E. (1996). Consolidation in human motor memory. *Nature, 382*(6588), 252–255.

Bullinger, H.-J., Breining, R., & Braun, M. (2001). Virtual reality for industrial engineering: Applications for immersive virtual environments. In G. Salvendy (Ed.), *Handbook of industrial engineering: Technology and operations management* (3rd ed.; pp. 2496–2520). New York: Wiley.

Bungert, C. (2005). *HMD/headset/VR-helmet comparison chart*. Retrieved November 1, 2006, from http://www.stereo3d.com/hmd.htm

Burdea, G. (1996). *Force and touch feedback for virtual reality*. New York: Wiley.

Burdea, G. C., & Coiffet, P. (2003). *Virtual Reality Technology* (2nd ed.). Hoboken, NJ: Wiley.

Butler, R. A. (1987). An analysis of the monaural displacement of sound in space. *Perception & Psychophysics, 41*, 1–7.

Calvert, S. L. (2002). The social impact of virtual environment technology. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 663–680). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Calvert, S. L., & Tan, S. L. (1994). Impact of virtual reality on young adult's physiological arousal and aggressive thoughts: Interaction versus observation. *Journal of Applied Developmental Psychology, 15*, 125–139.

Carretta, T. R., & Ree, M. J. (1993). Basic attributes test (BAT): Psychometric equating of a computer-based test. *International Journal of Aviation Psychology, 3*, 189–201.

Carretta, T. R., & Ree, M. J. (1995). Air Force Officer Qualifying Test validity for predicting pilot training performance. *Journal of Business and Psychology, 9*, 379–388.

Carson, D. (2000a). Environmental storytelling, part 1: Creating immersive 3D worlds using lessons learned from the theme park industry. Retrieved March 15, 2007, from http://www.gamasutra.com/features/20000301/carson_pfv.htm

Carson, D. (2000b). Environmental storytelling, part 2: Bringing theme park environment design techniques lessons to the virtual world. Retrieved March 15, 2007, from http://www.gamasutra.com/features/20000405/carson_pfv.htm

Chen, J. L., & Stanney, K. M. (1999). A theoretical model of wayfinding in virtual environments: Proposed strategies for navigational aiding. *Presence: Teleoperators and Virtual Environments, 8*(6), 671–685.

Chua, P. T., Crivella, R., Daly, B., Hu, N., Schaaf, R., Ventura, D., et al. (2003). Training for physical tasks in virtual environments:

Tai Chi. *Proceedings of IEEE Virtual Reality Conference 2003* (pp. 87–97). Los Angeles, CA, March 22–26, 2003.

Church, D. (1999, August). Formal abstract design tools. *Games Developer Magazine,* 44–50. Retrieved October 1, 2006, from http://www.gamasutra.com/features/19990716/design_tools_02.htm

Cohen, M. (1992). Integrating graphic and audio windows. *Presence: Teleoperators and Virtual Environments, 1*(4), 468–481.

Cohn, J., DiZio, P., & Lackner J. R. (2000). Reaching during virtual rotation: Context-specific compensation for expected Coriolis forces. *Journal of Neurophysiology, 83*(6), 3230–3240.

Cruz-Neira, C., Sandin, D. J., & DeFanti, T. A. (1993). Surround-screen projection-based virtual reality: The design and implementation of the CAVE. *ACM Computer Graphics, 27*(2), 135–142.

Crystal Rivers Engineering (1995). Snapshot: HRTF Measurement System. Groveland, CA.

Darken, R. P., & Peterson, B. (2002). Spatial orientation, wayfinding, and representation. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 493–518). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Davies, R. C. (2002). Applications of system design using virtual environments. In K.M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 1079–1100). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

DiZio, P., & Lackner, J. R. (1995). Motor adaptation to Coriolis force perturbations of reaching movements: Endpoint but not trajectory adaptation transfers to the non-exposed arm. *Journal of Neurophysiology, 74*(4), 1787–1792.

Draper, M. H., Prothero, J. D., & Viirre, E. S. (1997). Physiological adaptations to virtual interfaces: Results of initial explorations. *Proceedings of the Human Factors & Ergonomics Society 41st Annual Meeting* (pp. 1393). Santa Monica, CA: Human Factors & Ergonomics Society.

Durlach, B. N. I., & Mavor, A. S. (1995). *Virtual Reality: Scientific and technological challenges*. Washington, DC: National Academy Press.

Emmelkamp, P. M. G., Bruynzeel, M., Drost, L., & van der Mast, C. A. P. G., (2001). Virtual reality treatment in acrophobia: A comparison with exposure in vivo. *Cyberpsychology & Behavior, 4*(3), 335–339.

Favalora, G. E., Napoli, J., Hall, D. M., Dorval, R. K., Giovinco, M. G., Richmond, M. J., et al. (2003). 100 million-voxel volumetric display. *SPIE Proceedings of the 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 4297*, 227–235.

Fencott, C. (1999). Content and creativity in virtual environment design. *Proceedings of Virtual Systems and Multimedia '99*, University of Abertay Dundee, Scotland, 308-317. Retrieved March 15, 2007, from http://www.cs.york.ac.uk/hci/kings_manor_workshops/UCDIVE/fencott.pdf

Figueroa, P., Bischof, W. F., Boulanger, P., & Hoover, H. J. (2005). Efficient comparison of platform alternatives in interactive virtual reality applications. *International Journal of Human-Computer Studies, 62*(1), 73–103.

Fouad, H. (2004). Ambient synthesis with random sound fields. In K. Greenebaum (Ed.), *Audio anecdotes: Tools, tips, and techniques for digital audio*. Natick, MA: A K Peters.

Fouad, H., & Ballas, J. (2000, April). An extensible toolkit for creating virtual sonic environments. Paper presented at the *International Conference on Auditory Displays, ICAD 2000*, Atlanta, GA.

Foxlin, E. (2002). Motion tracking requirements and technologies. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 163–210). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Gabbard, J. L., Hix, D., & Swan, J. E. II (1999). User-centered design and evaluation of virtual environments. *IEEE Computer Graphics and Applications*, *19*(6), 51–59.

Gershon, J., Zimand, E., Lemos, R., Rothbaum, B. O., & Hodges, L. (2003). Use of virtual reality as a distractor for painful procedures in patients with pediatric cancer: a controlled study. *CyberPsychology and Behavior, 6*, 657–661.

Gopher, D., Weil, M., & Bareket, M. (1994) Transfer of skill from a computer game trainer to flight. *Human Factors, 36*(3), 387–405.

Gress, W., & Willkomm, B. (1996).Simulator-based test systems as a measure to improve the prognostic value of aircrew selection. *Selection and Training Advances in Aviation: Advisory Group for Aerospace Reasearch and Development Conference Proceedings, Prague, Czech Republic, 588*, 15-1–15-4.

Hahn, J. K., Fouad, H., Gritz, L., & Lee, J. W. (1998). Integrating sounds and motions in virtual environments. *Presence: Teleoperators and Virtual Environments, 7*(1), 67–77.

Hale, K. S., & Stanney, K. M. (2004). Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundation. *IEEE Computer Graphics and Applications, 24*(2), 33–39.

Halle, M. (1997, May). Autostereoscopic displays and computer graphics. *Computer Graphics, 31*(2), 58–62.

Hays, R. T., Castillo, E. R., Seamon, A. G., & Bradley, S. K. (1997). A virtual environment for submarine ship handling: perceptual and hardware trade-offs. In M. J. Chinni (Ed.), *Proceedings of the 1997 Simulation MultiConference: Military Government, and Aerospace Simulations* (Simulation Series Vol. 29, No. 4, pp. 217–222). San Diego, CA: The Society for Computer Simulation International.

Heirich, A., & Arvo, J. (1997). Scalable Monte Carlo image synthesis [Special issue on Parallel Graphics & Visualization]. *Parallel Computing, 23*(7), 845–859.

Hettinger, L. J. (2002). Illusory self-motion in virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 471–492). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Hix, D., & Gabbard, J. L. (2002).Usability engineering of virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 681–699). Mahwah, NJ: Lawrence Erlbaum Associates.

Holden, M. (2005). Virtual environments for motor rehabilitation: A review. *CyberPsychology & Behavior, 8*(3), 187–211.

Holden, M., Todorov, E., Callahan, J., & Bizzi, E. (1999). Virtual environment training improves motor performance in two patients with stroke: case report. *Neurology Report, 23*, 57–67.

Hollerbach, J. (2002). Locomotion interfaces. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 239–254). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Huang, X. D. (1998). Spoken language technology research at Microsoft. *The Journal of the Acoustical Society of America*, *103*(5), 2815–2816.

Isdale, J. (2000, April). Motion simulation. *VR News: April Tech Review*. Retrieved (date), from http://vr.isdale.com/vrTechReviews/MotionSim_Links_2000.htm

Isdale, J., Fencott, C., Heim, M., & Daly, L. (2002). Content design for virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 519–532). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Jones, L., Bowers, C. A., Washburn, D., Cortes, A., & Vijaya Satya, R. (2004). The effect of olfaction on immersion into virtual environments. In D. A. Vincenzi, M. Mouloua, & P. A. Hancock (Eds.), *Human performance, situation awareness and automation: Current research and trends* (Vol. 2; pp. 282–285). Mahwah, NJ: Lawrence Erlbaum.

Kalawsky, R. S. (1993). *The science of virtual reality and virtual environments*. Wokingham, England: Addison-Wesley.

Kallman, E. A. (1993). Ethical evaluation: A necessary element in virtual environment research. *Presence: Teleoperators and Virtual Environments, 2*(2), 143–146.

Kaur, K. (1999). *Designing virtual environments for usability*. Unpublished doctoral dissertation, City University, London.

Kennedy, R. S., Drexler, J. M., Jones, M. B., Compton, D. E., & Ordy, J. M. (2005). Quantifying human information processing (QHIP): Can practice effects alleviate bottlenecks? In D. K. McBride & D. Schmorrow (Eds.), *Quantifying human information processing* (pp. 63–122). Lanham, MD : Lexington Books.

Kennedy, R. S., & Fowlkes, J. E. (1992). Simulator sickness is polygenic and polysymptomatic: Implications for research. *International Journal of Aviation Psychology, 2*(1), 23–38.

Kennedy, R. S., & Graybiel, A. (1965). *The Dial test: A standardized procedure for the experimental production of canal sickness symptomatology in a rotating environment* (Rep. No. 113, NSAM 930). Pensacola, FL: Naval School of Aerospace Medicine.

Kennedy, R. S., Kennedy, K. E., & Bartlett, K. M. (2002). Virtual environments and products liability. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 543–554). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology, 3*(3), 203–220.

Kennedy, R. S., & Stanney, K. M. (1996). Virtual reality systems and products liability. *Journal of Medicine and Virtual Reality, 1*(2), 60–64.

Kennedy, R. S., Stanney, K. M., & Dunlap, W. P. (2000). Duration and exposure to virtual environments: Sickness curves during and across sessions. *Presence: Teleoperators and Virtual Environments, 9*(5), 466–475.

Kennedy, R. S., Stanney, K. M., Ordy, J. M., & Dunlap, W. P. (1997). Virtual reality effects produced by head-mounted display (HMD) on human eye-hand coordination, postural equilibrium, and symptoms of cybersickness. *Society for Neuroscience Abstracts, 23*, 772.

Kershner, J. (2002). Object-oriented scene management. *GameDev.net*. Retrieved October 15, 2006, from http://www.gamedev.net/reference/articles/article1812.asp

Kessler, G. D. (2002). Virtual environment models. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 255–276). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Knerr, B. W., Breaux, R., Goldberg, S. L., & Thurman, R. A. (2002). National defense. In K. M. Stanney (Ed.), *Handbook of Virtual environments: Design, implementation, and applications* (pp. 857–872). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Lackner, J. R., & DiZio, P. (1994). Rapid adaptation to Coriolis force perturbations of arm trajectory. *Journal of Neurophysiology, 72*(1), 299–313.

Lawson, B. D., Sides, S. A., & Hickinbotham, K. A. (2002). User requirements for perceiving body acceleration. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 135–161). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Majumder, A. (2003). A practical framework to achieve perceptually seamless multi-projector displays. Unpublished doctoral dissertation, University of North Carolina at Chapel Hill. Retrieved March 15, 2007, from http://www.cs.unc.edu/~welch/media/pdf/dissertation_majumder.pdf

Massimino, M., & Sheridan, T. (1993). Sensory substitution for force feedback in teleoperation. *Presence: Teleoperators and Virtual Environments, 2*(4), 344–352.

Martinussen, M. (1996). Psychological measures as predictors of pilot performance:A meta-analysis. *International Journal of Aviation Psychology, 6*, 1–20.

May, J. G., & Badcock, D. R. (2002). Vision and virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 29–64). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

McCauley, M. E., & Sharkey, T. J. (1992). Cybersickness: Perception of self-motion in virtual environments. *Presence: Teleoperators and virtual environments, 1*(3), 311–318.

McCauley-Bell, P. R. (2002). Ergonomics in virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 807–826). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Menzies, D. (2002, July). Scene management for modelled audio objects in interactive worlds. *Proceedings of the 8th International Conference on Auditory Displays*, Kyoto, Japan.

Merritt, E. A., & Bacon, D. J. (1997). Raster3D: Photorealistic molecular graphics. *Methods in Enzymology, 277*, 505–524.

Milham, L. M., Hale, K., Stanney, K., Cohn, J., Darken, R., & Sullivan, J. (2004). When is VE training effective? A framework and two case studies. Poster presented at *The 48th Annual Human Factors and Ergonomics Society Meeting* (pp. 2592–2595). New Orleans, LA, September 20–24, 2004.

Moline, J. (1995). *Virtual Environments for Health Care. White Paper for the Advanced Technology Program* (*ATP*). Retrieved September 15, 2006, from the National Institute of Standards and Technology website: http://www.itl.nist.gov/iaui/ovrt/projects/health/vr-envir.htm.

Morris, C. S., & Tarr, R. W. (2002, March). Templates for selecting PC-based synthetic environments for application to human performance enhancement and training. *Proceedings of IEEE Virtual Reality 2002 Conference*, Orlando, FL, 109–115 .

Mulgund, S., Stokes, J., Turieo, M., & Devine, M. (2002). *Human/machine interface modalities for soldier systems technologies* (Final Report No. 71950-00). Cambridge, MA: TIAX, LLC.

Murray, J. H. (1997). *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* New York: The Free Press.

Nguyen, L. K., Cohn, J. V., Mead, A., Helmick, J., & Patrey, J. (2001). Real-time virtual environment applications for military maritime training. In M. J. Smith, G. Salvendy, D. Harris, & R. J. Koubek (Eds), *Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality* (Vol. 1 of the Proceedings of HCI International 2001) (pp. XXX–YYY). Mahwah, NJ: Lawrence Erlbaum.

North, M. M., North, S. M., & Coble, J. R. (2002). Virtual reality therapy: An effective treatment for psychological disorders. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 1065–1078). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological Review, 56*, 132–143

Popescu, G. V., Burdea, G. C., & Trefftz, H. (2002). Multimodal interaction modeling. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 435–454). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Pountain, D. (1996, July). VR meets reality: Virtual reality strengthens the link between people and computers in mainstream applications. *Byte Magazine*. Retrieved October 20, 2006, from http://www.byte.com/art/9607/sec7/art5.htm.

Povenmire, H. K., & Roscoe, S. N. (1973). Incremental transfer effectiveness of a ground-based general aviation trainer. *Human Factors, 15*(6), 534–542.

Rizzo, A. A., Buckwalter, G. J., & van der Zaag, C. (2002). Virtual environment applications in clinical neuropsychology. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 93–116). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Rolland, J. P., Biocca, F. A., Barlow, T., & Kancherla, A. (1995). Quantification of adaptation to virtual-eye location in see-thru head-mounted displays. *IEEE Virtual Reality Annual International Symposium '95* (pp. 56–66). Los Alimitos: CA: IEEE Computer Society Press.

Roscoe, S. N. (1982). *Aviation Psychology.* Ames, IA: Iowa State University Press.

Rothbaum, B. O., Hodges, L., Watson, B. A., Kessler, G. D., & Opdyke, D. (1996). Virtual reality exposure therapy in the treatment of fear of flying: A case report. *Behaviour Research & Therapy, 34*, 477–481.

Sadowski, W., & Stanney, K. (2002). Presence in virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 791–806). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Sanders, P. (1997, September/October). Simulation Based Acquisition: An effective, affordable mechanism for fielding complex technologies. *Program Manager,* 72–77.

Satava, R., & Jones, S. B. (2002). Medical applications of virtual environments. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 93–116). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Schmorrow, D., Stanney, K. M., Wilson, G., & Young, P. (2005). Augmented cognition in human-system interaction. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (3rd ed.). New York: John Wiley.

Sheridan, T. B. (1993). My anxieties about virtual environments. *Presence: Teleoperators and Virtual Environments, 2*(2), 141–142.

Shilling, R. D., & Shinn-Cunningham, B. (2002). Virtual auditory displays. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 65–92). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Siem, F. M., Carretta, T. R., & Mercatante, T. A. (1988). *Personality, attitudes, and pilot training performance: Preliminary analysis* (Tech. Report. N. AFHRL-TP-87-62). Brooks Air Force Base, TX: AFHRL, Manpower and Personnel Division.

Singhal, S., & Zyda, M. (1999). *Networked Virtual Environments—Design and Implementation* (SIGGRAPH Series). New York: ACM Press Books.

Slater, M., & Steed, A. (2000). A virtual presence counter. *Presence: Teleoperators and Virtual Environments, 9*(5), 413–434.

Stanney, K. M., Graeber, D. A., & Kennedy, R. S. (2005). Virtual environment usage protocols. In W. Karwowski (ed), *Handbook of standards and guidelines in ergonomics and human factors* (pp. 381–398). Mahwah, NJ: Lawrence Erlbaum.

Stanney, K. M., & Kennedy, R. S. (1998, October). Aftereffects from virtual environment exposure: How long do they last? *Proceedings of the 42nd Annual Human Factors and Ergonomics Society Meeting,* Chicago, 1476–1480.

Stanney, K. M., Kingdon, K., Nahmens, I., & Kennedy, R. S. (2003). What to expect from immersive virtual environment exposure: Influences of gender, body mass index, and past experience. *Human Factors, 45*(3), 504–522.

Stanney, K. M., Mollaghasemi, M., & Reeves, L. (2000). *Development of MAUVE, the Multi-Criteria Assessment of Usability for Virtual Environments System* (Final Report, Contract No. N61339-99-C-0098). Orlando, FL: Naval Air Warfare Center, Training Systems Division, 8/00.

Stanney, K. M., Mourant, R., & Kennedy, R. S. (1998). Human factors issues in virtual environments: A review of the literature. *Presence: Teleoperators and Virtual Environments, 7*(4), 327–351.

Stanney, K. M., Salvendy, G., Deisigner, J., DiZio, P., Ellis, S., Ellison, E., et al. (1998). *Aftereffects and sense of presence in virtual environments: Formulation of a research and development agenda* (Report sponsored by the Life Sciences Division at NASA Headquarters). *International Journal of Human-Computer Interaction, 10*(2), 135–187.

Stanney, K., Samman, S., Reeves, L., Hale, K., Buff, W., Bowers, C., et al. (2004). A paradigm shift in interactive computing: Deriving multimodal design principles from behavioral and neurological foundations. *International Journal of Human-Computer Interaction, 17*(2), 229–257.

Stanney, K. M., & Zyda, M. (2002). Virtual environments in the 21st century. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 1–14). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Stoffregen, T., Bardy, B. G., Smart, L. J., & Pagulayan, R. (2003). On the nature and evaluation of fidelity in virtual environments. In L. J. Hettinger & M. W. Haas (Eds.), *Virtual and adaptive environments: Applications, implications, and human performance issues.* (pp. 111–128). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Stone, R. (2002). Applications of virtual environments: An overview. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications*. (pp. 827–856). Mahwah: NJ: Lawrence Erlbaum Associates.

Storms, R. L. (2002). Auditory-visual cross-modality interaction and illusions. In K. M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 455–470). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Strickland, D., Hodges, L., North, M., & Weghorst, S. (1997). Overcoming phobias by virtual exposure. *Communications of the ACM, 40*(8), 34–39.

Swartz, K. O. (2003). *Virtual environment usability assessment methods based on a framework of usability characteristics.* Unpublished master's thesis. Virginia Polytechnic Institute and State University, Blacksburg.

Thacker, P. D. (2003). Fake worlds offer real medicine: Virtual Reality finding a role in treatment and training. *Journal of the American Medical Association, 290*(16), 2107–2112.

Tharp, V., Burns, M., & Moskowitz, H. (1981). *Development and field test of psychophysical tests for DWI arrest* (Department of Transportation Final Report, ODT HS 805 864). Washington, DC: DOT.

Thorndike, E. L., & Woodworth, R. S. (1901). The influence of improvement of one mental function upon the efficiency of the other functions. *Physiological Review, 8*(3), 247–261.

Tredennick, N., & Shimamoto, B. (2005). The vision of Microvision. *Glider Technology Report.* Vol 10, No 10. Retrieved March 15, 2007, from http://www.microvision.com/documents/GTROctober05.pdf

Turk, M. (2002). Gesture recognition. In K.M. Stanney (Ed.), *Handbook of virtual environments: Design, implementation, and applications* (pp. 223–238). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Ultimate 3D Links (2007). *Commercial 3D software.* Retrieved March 15, 2007, from http://www.3dlinks.com/links.cfm?categoryid=1&subcategoryid=1#.

Vince, J. (2004). *Introduction to virtual reality* (2nd ed.). Berlin: Springer-Verlag.

Welch, R. B. (1978). *Perceptual modification: Adapting to altered sensory environments*. New York: Academic Press.

Welch, R. B. (1997). The presence of aftereffects. In G. Salvendy, M. Smith, & R. Koubek (Eds.), *Design of computing systems: Cognitive considerations* (pp. 273–276). Amsterdam, Netherlands: Elsevier Science Publishers, San Francisco, CA, August 24–29.

Wickens, C. D., & Hollands, J.G. (2000) *Engineering psychology and human performance* (3rd edition). New Jersey: Prentice Hall.

Witmer, B., & Singer, M. (1998). Measuring presence in virtual environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments, 7*(3), 225–240.

Zittel, R. C. (2001, Summer). The reality of simulation-based acquisition-And an example of U.S. military implementation. *Acquisition Review Quarterly,* 121–132.

·32·

HUMAN-COMPUTER INTERACTION VIEWED FROM THE INTERSECTION OF PRIVACY, SECURITY, AND TRUST

John Karat, Clare-Marie Karat, and Carolyn Brodie
IBM T.J. *Watson Research Center*

Introduction **640**
HCI and Usability in the Privacy, Security, and Trust Domains **641**
Views on Privacy 641
HCI Studies of Privacy 642
Anonymity 643
Personalization 643
End user views of privacy 643
Organizational requirements in managing privacy 644
Summary of Findings and Recommendations 645
Views on Security 646
Usability of Security Software and Mechanisms 646
Control Over Security 647
Key Topic Areas for HCI in Security 647
Secure design and administration 647
Authentication mechanisms 648
Passwords 648
Biometrics 649
Other security approaches 650
Usability Challenges Specific to Security—Attack Defense 650
Summary of Findings and Recommendations 651
Views on Trust 651
Factors Influencing Trust 651
Trust and Risk 652
Trust and Personalization 653
Summary of Findings and Recommendations 653
Conclusions and Directions **654**
References **655**

INTRODUCTION

What do we mean when we say that we trust a computer system? While trust is a complex concept with many different meanings in behavioral science literature, it has a specific meaning when applied to a person's interaction with a technology system. Trust in a system is the extent to which a system behaves as the user expects. Put another way, it is a measure of how much users believe that the system will do what they ask it to do without causing any harm. This view suggests that people can enter an interaction with a system with some ideas about what the interaction will involve, and that their trust in the system can either increase or decrease with experience. Trust becomes important, particularly for people or organizations that rely on people wanting to use their websites or products, because without trust in a system, people will find other ways to conduct business. Parts of the expectations people have about systems are related to the intended function or purpose of the system. They trust that ATMs will correctly distribute money and debit their accounts appropriately. They expect that merchandise ordered through websites will arrive safely at their homes. However, there are also expectations about the handling of information they exchange with the systems. They assume that transactions are secure—that they know with whom they are dealing and that the information reaches only the intended target. They also assume that their privacy is respected—that promises the owners of the system make about how the information will be used or shared are kept. This intersection of trust, security, and privacy is becoming increasingly important, as information technology becomes a more pervasive part of our lives.

As we will discuss below, there can be many aspects and influences on the trust placed by a user in a computer system. We will not try to address them all in this chapter, but will focus on two important contributing factors associated with the risks of harm one might have in interacting with a system. When people interact with systems, they assume that doing so will not cause them any harm. They expect, that is they trust, that the interactions are truthful and without hidden consequences. If they interact with websites, they assume that it represents communication with a specific person or organization. If they disclose information to the websites, they assume that the disclosures will only reach the intended targets and that only appropriate use will be made of the information.

We will consider security as the concept associated with how well a system can protect information it contains. Many factors are associated with security in systems that interact with humans. Perhaps the most important of these is authentication—how a system and a user can be confident of each other's identities. If users and systems have confidence that they know the identities of the others, we generally assume that neither will attempt to do harm to the other. While research in system security is often aimed at minimizing risks associated with malicious attacks on systems, for HCI we will focus on the tradeoffs between rigorous authentication and ease of interaction.

The concept of privacy goes beyond security to examine how well the use of information that the system acquires about a user conforms to the explicit or implicit assumptions regarding that use associated with the personal information. We would like to draw an important distinction when discussing privacy from an HCI view. In general, studies have looked at privacy from two different perspectives. From an end-user perspective, privacy can be considered as restricting access to personal information, or it can be viewed as controlling use of personal information. In the former, the user expresses a wish to be left alone; in the latter the wish is to use information according to expressed wishes. While much has been said about the end of privacy in the pervasive computing world—meaning that so much is known about each of us that it is futile to worry about privacy—this has really been an attempt to highlight the difficulty individuals face in trying to remain anonymous (the first consideration above). While electronic surveillance is increasingly common, this does not mean that people have to give up their rights to control the use of information collected about them. It is our assumption that collection of more and more data about us is a trend that will continue. However, we also assume that legislation will support people's rights to ensure that appropriate use is made of data collected. In considering privacy, we will generally assume that security in a system is adequate. Thus, we will view data protection failures associated with unauthorized access to information to be security failures or breaches, and those associated with noncompliance to stated privacy policies to be privacy breaches. We will explore the privacy issues in more detail below. For the purposes of this chapter, a simple but useful definition of privacy is "the ability of individuals to control the terms under which their personal information is acquired and used" (Culnan, 2000, p. 21).

In summary, security involves technology to ensure that information is appropriately protected. It involves users in that security features such as passwords for access control or encryption to prevent unintended disclosure, often place requirements on users in order to function correctly. Password schemes must be hard to guess but should be easy to remember. Encryption mechanisms must prevent unintended decryption but be transparent (or nearly so) to intended senders and recipients. Privacy involves mechanisms to support compliance with some basic principles (e.g., OECD, 1980; U.S. Fair Credit Reporting Act of 1978, 1978). These suggest that people should be informed about information collection, told in advance what will be done with their information, and given a reasonable opportunity to approve of such use of information. Trust is seen as increasing when security and privacy are provided for. Without trust, it is perceived that people will be less likely to use systems.

As a part of our research, we identified connections between the concepts privacy, security, and trust as they applied to interaction with organizations that collect personal information (PI). This work (Karat, J., Karat, C., Brodie, & Feng, 2005) employed a Contextual Design method that enabled us to identify themes present in data and to understand the relationships among them. One theme involved the relationship of privacy to other concepts. Many of the participants we talked with discussed how privacy related to other concepts such as security, personalization, trust, and education. Fig. 32.1 below shows how each of these concepts relates to privacy. Good security enables and is a building block for privacy, and the interviewees have noted that, as they focus on managing privacy in their organizations, they have found ways to enhance their security as well. The trust that data subjects have in an organization is paramount, and

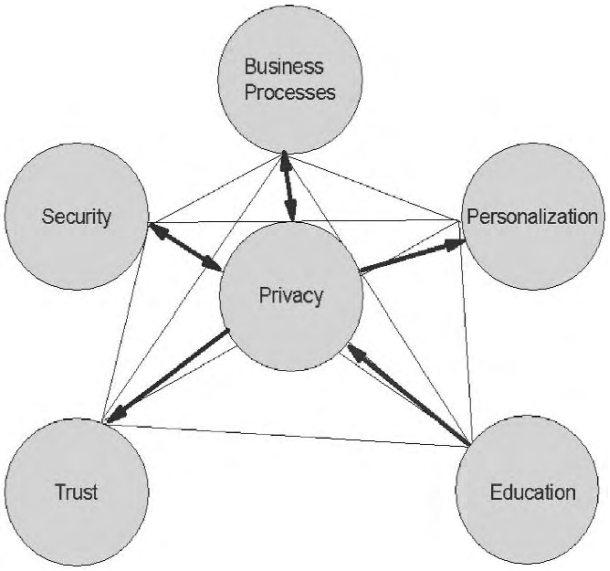


FIGURE 32.1. The relationships between privacy and other concepts.

providing privacy protection is a critical means of ensuring that trust. Privacy education is important for both the organizations and their external users. Many interviewees stated that privacy education of their employees is an ongoing priority now and that the privacy of data subjects' PI cannot be effectively protected until employees (the data users) believe it is important and follow through in their daily actions. Many have initiated training programs and have frequently asked questions (FAQs) available for employees to access to understand how to handle PI correctly. The interviewees in the current research study stated that their customers would only provide PI on the condition that it would be protected and not misused, and this is a critical element in personalization. Privacy is woven into organizations' business processes. Interviewees stated that they are reexamining their business processes in order to protect the personally identifying information (PII) of their employees, customers, constituents, and patients. They are finding redundancies in the collection of PII and are realizing financial benefits from streamlining their processes privacy management.

The privacy and security functionality of the applications that users experience must also be usable to gain the user's trust. There are unique challenges in making these capabilities usable. We will explore these issues before discussing the research in privacy, security, and trust.

HCI AND USABILITY IN THE PRIVACY, SECURITY, AND TRUST DOMAINS

While the core usability goals of understanding the user, their tasks, and the context of use apply across a wide range of domains, unique aspects of privacy and security present challenges and opportunities when designing security and privacy functionality. First, a key issue to consider is that the use of security and privacy solutions is not the user's main goal. Users value and want security and privacy functionality, and they are not likely to trust systems that do not provide them, but they regard them as only secondary to completing their primary tasks (e.g., completing an online banking transaction, ordering medications). The central conflict in the minds of many is that security mechanisms are seen as making operations harder, while usability is focused on making operations easier. The user would like the solutions to be as transparent as possible, but users do want to be in control and understand what is occurring. Therein lies the HCI challenge. Thus, the display of information and the interaction methods related to security and privacy solutions need to be accessible when desired by the user.

Second, as more of people's interactions in daily life involve the use of computing technology and sensitive information, disparate types of users must be accommodated. Security solutions in particular have historically been designed with a highly trained technical user in mind. The user community has broadened extensively as organizational business processes have come to include new types of roles and users in the security and privacy area. Many compliance and policy roles in organizations are handled by legal and business process experts who have limited technical skills. Moreover, the end user base includes virtually everyone in the population. The functionality provided by the system for people with different roles must accommodate the skill sets of each. Security and privacy are requirements for doing business as an organization and must be done well, or the organization may lose the user as a customer or worse.

Third, the risk of the negative impact of usability problems is higher for security and privacy applications than for many other types of systems. Although complexity is at the very heart of many security and privacy solutions, from an HCI point of view that complexity is really the enemy of the success of security and privacy. If the system is so complex that the various users groups (e.g., technical users, business users, and end users) cannot understand it, then costly errors will occur. There is a saying in the HCI field that if the user cannot understand functionality, it doesn't exist. In the case of security and privacy, badly designed functionality may put users at more risk than if they used less sophisticated solutions. The increased risk of errors in this domain provides an even greater incentive to include HCI work in system research and development. The user issues in the domain provide unique technical challenges to architects to design the solutions to be simple and effective.

Fourth, users will need to be able to easily update security and privacy solutions to accommodate frequent changes in legislation and regulations. Additionally, different domains (e.g., health care, banking, government) and geographies will have unique requirements. Systems must be designed to enable easy and effective updates to them. While this list is not exhaustive, these challenges provide a unique focus and strong incentive to include HCI in the system lifecycle. We discuss the valuable role that HCI can play in more detail below.

Views on Privacy

The rapid advancement of the use of information technology in industry, government, and academia makes it much easier to

collect, transfer, and store PI around the world. This raises challenging questions and problems regarding the use and protection of PI (Kobsa, 2002). Questions of who has what rights to information about us for what purposes become more important as we move toward a world in which it is technically possible to know just about anything about just about anyone. As Adams and Sasse (2001, p. 49) stated, "Most invasions of privacy are not intentional but due to designers' inability to anticipate how this data could be used, by whom, and how this might affect users." Deciding how we are to design privacy considerations in technology for the future includes philosophical, legal, and practical dimensions—any or all of which can be considered as within the domain of the field of HCI.

We provide a description of our scope in addressing this area, since privacy can and does mean different things to different people. We are primarily focused on a view of privacy as the right of an individual to control personal information use rather than as the right to individual isolation (National Research Council, 2003; OECD, 1980; U.S. Fair and Accurate Credit Transaction Act, 2003). Organizations commonly provide a description of what kind of information they will collect and how they will use it in privacy policies. In some areas (e.g., the collection and use of health care information in the United States or movement of personal information across national boundaries in Europe), such policies can be required, though the content of the policy is not generally specified in legislation. While there has been considerable consensus around a set of high-level privacy principles for information technology (OECD, 1980), we do not think it is likely that a single privacy policy can be created to address all information privacy needs. For example, there will likely be considerable differences in privacy legislation in different regions of the world (Manny, 2003). Similarly, organizations in different fields (e.g., healthcare, banking, government) need to tailor policies to their domains and needs (Ball, 2003; Baumer, Earp, & Payton, 2000). While we will focus on privacy policy, we acknowledge that privacy is not entirely about "setting rules and enforcing them" (Palen & Dourish, 2002). To implement privacy within an organization, the coordination of people, business processes, and technology is required. Still we do believe that such policies are essential when interacting with technology and organizations in that they enable people to better understand the boundaries between public and private information and technology (Altman, 1975).

It is interesting to note that while privacy policies are not new to most organizations, very little has been done to implement the policies through technology (Smith, 1993). There are emerging standards for privacy policies on websites (Cranor, 2002), but these address machine-readable policy content without specifying how the policy might be created or implemented. The reality is that there is very little capability to have technology actually implement access and disclosure limitations that we might expect from a policy statement like "We will not share your information with a third party without your consent." The emerging focus is on how organizations could create a wide range of policies, and how technology might enable the policies to be enforced and audited for compliance. J. Karat, C. Brodie, C. Karat, and J. Feng (2005) focused on technology to enable usable privacy policy authoring and enforcement, rather than trying to directly address what privacy rights people should have (e.g., Warren & Brandeis, 1890) or how to deidentify information such as video stored in systems (e.g., Senior, Pankanti, Hampapur, Brown, Tiann, & Ekin, 2004). We think privacy is an important social issue and that technology can enable flexible, reliable and accountable privacy policy (e.g., be privacy enabling) and not just be a force which reduces individual rights.

The current situation with respect to privacy in organizations is described in a Forrester report (Hagen, 2000). This research reveals a mismatch between consumer demands for privacy and enterprise practices in industry. According to this report, although customer concerns about privacy remain high, the majority of executives (58%) believe privacy issues are addressed extremely well by their companies. Most executives do not know whether their customers check the privacy policies or not and few see the need to enhance their privacy practices. Research in the Asia-Pacific region complements these results (Office of the Federal Privacy Commissioner of Australia, 2000). With these results in mind, we suggest that privacy protection must extend across the network into enterprise processes and that there is a need to audit data collection and sharing mechanisms. We agree that technology design generally does reflect concerns of society in general (Ackerman & Mainwaring, 2005), and believe that we are experiencing a shift toward a greater concern for privacy in IT design.

Central to our view of privacy is the notion that the parties involved in information exchanges have implicit or explicit policies with regard to the use of the information. This applies both to the person who the information is about and to the organization collecting and using the information. In the privacy literature on organizations, while there has been some attention given to the generally implicit policies of end users whose PI is being collected and used (often called data-subjects), it is mostly the policies of the organization collecting the information that has been given attention. Smith (1993) described such organizational policies and noted the lack of technology in enforcing the policies. He described the rather unstructured ways in which organizations develop privacy policy—a characterization that has changed little in the nearly 10 years since his research was published in spite of the increased legislation of the past few years. In our work, we try to address both the needs of data subjects and organizations by addressing the gap between policy and practice.

HCI *Studies of* Privacy

Rapid advances in technology are pushing us closer and closer to Mark Weiser's (1991) vision of a ubiquitous computing world. Technology has permeated every facet of our daily lives—from enabling us to shop online with a few clicks of the mouse, to recommending interest-appropriate movies (Good, et al., 1999), to helping us share documents with colleagues on the team (e.g., Horstmann & Bentley, 1997), to allowing us to estimate when a collaborator may be available for communication (e.g., Begole, Tang, Smith, & Yankelovich, 2002), to reminding us of relevant tasks based on our current context (e.g., Dey & Abowd, 2000), and so on. The benefits of these technological developments promise to improve quality and experience of life by making it more convenient and efficient. In order to effectively provide these benefits, systems need to capture, store,

and analyze information that may be deemed sensitive by the individuals concerned. The quantity of information varies from system to system, and the associated sensitivity can vary from individual to individual and context to context. As a result, to achieve their potential, such systems must first overcome various social hurdles. In particular, user perceptions regarding potential violations of privacy have emerged as a key factor that affects technological acceptance and adoption (Herbsleb, Atkins, Boyer, Handel, & Finholt, 2002; Want, Hopper, Falcão, & Gibbons, 1992). For example, the ambitious Total Information Awareness (TIA) initiative proposed by the U.S. government was abandoned before proceeding beyond the planning stages due to its inability to address satisfactorily privacy fears raised by citizens concerned with civil liberties (Associated Press, 2003). As rampant increases in viruses, worms, spam, spyware, and adware threaten to erode user confidence, it has become increasingly important that a system empower users to appropriately manage privacy.

The motivation in studying these issues within the field of HCI lies in the interest in designing technological systems with privacy in mind at the outset. While researchers have acknowledged the existence of privacy concerns in a wide variety of technological domains (e.g., e-mail; Bellotti, 1996; e-commerce; e.g., Ackerman, Cranor, & Reagle, J. 1999; media spaces; e.g., Mantei, Baecker, Sellen, Buxton, Milligan, & Wellman, 1991; data mining; e.g., Thuraisingham, 2002; homes of the future; e.g., Meyer & Rakotonirainy, 2003; and so on), designing effective solutions to address these concerns is challenging. One of the reasons is that relatively few empirical studies have been conducted with the sole aim of studying privacy issues. In addition, empirical investigations of privacy pose numerous methodological challenges. To start with, privacy has proved hard to define due to its highly nuanced and context-dependent nature. Thus, individual differences in perceptions and interpretations of privacy could lead to researcher-introduced bias. For example, in a review of 23 privacy surveys, Harper and Singleton (2001) pointed out that surveys may suffer from manipulative questioning on the one hand, and that unprompted surveys may reveal very little privacy concern on the other hand. It is also possible that certain privacy issues remain undetected because the researchers did not recognize them as such. Moreover, cultural differences in expectations and behaviors regarding these issues tend to be quite profound (Milberg, Burke, Smith, & Kallman, 1995) making it difficult to generalize findings across cultures, or to study settings that involve individuals from multiple cultures. Differences in privacy laws in different countries could make it difficult to isolate actual intention from mere legal compliance. Further, methodologies for studying privacy may themselves be deemed too privacy invasive, causing users to deviate from normal practice and to withhold revealing sensitive aspects. As a result, relying on self-reported attitudes and behavior alone may not provide a valid view of normal practices (Spiekermann, Grossklags, & Berendt, 2001).

With this overview of the privacy and HCI area as context, we will examine some key research topic areas in more depth. These areas will include anonymity, personalization, and end-user views of privacy, including privacy policies, the Platform for Privacy Preferences (P3P), Peer-to-Peer (P2P), ubiquitous computing, and electronic voting. We will conclude this review with a discussion of the views and requirements of end users within organizations who are responsible for managing the personal data held by the organization.

Anonymity. Anonymity and anonymization in information technology includes the ability of a user to maintain privacy while completing transactions on a network and the ability for a user to keep the data they provide from identifying them personally. In researching anonymizing networks, Dingledine and Mathewson (2005) concluded that, in order for users to be able to preserve their privacy on networks by completing transactions without revealing communication partners, it is critical that the system chosen is usable so that other users can successfully employ the system as well. Anonymizing networks are successful at hiding users among users. An eavesdropper may be able to determine who is using the network but cannot identify who completes a particular transaction and the larger the group of users on the network, the more anonymous the participants become. The catch phrase in this area is "anonmymity loves company" (Reiter & Rubin, 1998).

Malin and Sweeney (2004; Sweeney, 2002) demonstrated that minimal amounts of information believed to be anonymous can be used to personally identify an individual. Since data are aggregated from a myriad of databases in the networked world in which we live, users may falsely believe that they can remain anonymous by providing only minimal bits of information in transactions here and there over time. When the data are aggregated though, the person can be identified easily based on three minimal pieces of information. Users may not think about the types of data from different sources that might be combined and analyzed in order to identify them, and thus there is a false sense of privacy. This research also makes a gray area clear in terms of what data elements might be labeled "PI" versus "PII"—it depends on the other data available in context.

Personalization. J. Karat, C. Karat, and Brodie (2004) stated that the user experience in HCI is different as the systems with which users interact become aware of the context of the interaction. Personalizing interaction involves the use of information about the user to alter the content presented to provide value to the user. In their empirical studies, J. Karat, C. Karat, and Brodie found that the most critical element in the willingness of users to adopt and use personalization systems was user control of personal data. Users want to be in control of their privacy and appreciate the benefits of personalization when they have control over the use of their personal data.

Cranor (2004) complemented this line of research with an analysis of privacy risks associated with personalization and presented a set of design guidelines to reduce privacy risks in personalization systems. Telzrow and Kobsa (2004) completed a meta-analysis of 30 consumer studies of user privacy preferences regarding personalization systems and found that consumer demands and current practice diverge significantly regarding control of personal information. The vast majority of businesses allow neither control over what information is stored nor ability to access it for verification, correction, or updates.

End user views of privacy. Researchers have conducted many studies on end user preferences regarding privacy on the

web and the HCI factors that are necessary to satisfy the user's desire for privacy. Jensen and Potts (2004) found that, while most surveys of user concerns about privacy show high rates of behaviors such as reading the privacy policy or taking concrete actions to protect their privacy, informal analysis of log-file data suggests that the actual rates of these user behaviors are much lower. Jensen, Potts, and Jensen (2005) found that users place inappropriate trust in the presence of trust indicators such as the TRUSTe mark or a privacy policy on the website, assuming quality in the presence of these trust indicators rather than understanding that the level of their trust should be dependent on the content of these policies. The users were willing to divulge personal information when it was not warranted. Jensen, Potts, and Jensen recommended that HCI professionals work to increase awareness of the privacy issues related to the policy issues associated with the trust indicators, as unscrupulous online vendors could use the trust indicators to mislead users to accept privacy policies and divulge private information that they not would otherwise willingly do. Buffett, Fleming, Richter, Scott, and Fleming (2004) created a technique for enabling a user to compute the value of the consequences of exchanging personal information on the web. The paper provides a demonstration of the effectiveness of the technique in improving a user's expected utility in a simple privacy negotiation.

Cranor (2005) led the W3C standardization work in the area of the Platform for Privacy Preferences (P3P) policies. P3P policies provide end users with information about the privacy policies of a website before they interact with it. Cranor and a team at AT&T Labs designed and developed the Privacy Bird (Cranor, 2002), a browser-help object, that provided summary information about the agreement or lack thereof between the end user's privacy preferences and the privacy policy of the website with which the person was considering interacting. Research continues to improve the usability of the Privacy Bird in communicating to users about website privacy policies and in capturing user privacy preferences.

Peer-To-Peer (P2P) systems enable users to share easily files by downloading simultaneously from multiple sources and by sharing many different file types. Good and Krekelberg (2003) illustrated HCI problems with P2P file sharing systems such as KaZaA. KaZaA, the most popular and widely used in 2003, had an average of 120 million downloads worldwide and 3 million users online at any given time during the course of the study. Good and Krekelberg (2004) conducted a user study of KaZaA and found that a large proportion of users unwillingly shared personal and private files, leaving them at risk of being taken advantage of through unknown exposure of PI. The majority of the users in the study were unable to correctly determine the files they were sharing. Many thought they were sharing no files when they were actually sharing all of the files on their hard drives.

The area of ubiquitous computing presents many new capabilities for users in communicating with the outside world through a number of different mobile, pervasive devices and through many choices in terms of managing their privacy in these new computing contexts. A group of researchers conducted a series of research studies in this domain. In the first study, Lederer, Mankoff, and Dey (2003) conducted a questionnaire-based study on the relative importance of the inquirer and the situation in determining a user's decision regarding the preferred accuracy of PI disclosed through the ubiquitous device. They found that users were more likely to provide the same level of accuracy in the disclosure of PI to the same inquirer in different situations rather than to different inquirers in the same situation. In later research, Hong, Ng, Lederer, and Landay (2004) developed a privacy-risk model for designing privacy-sensitive ubiquitous computing systems and presented two case studies that illustrated the use of the privacy-risk model in the design of applications. The goal was to refine the concepts of privacy to concrete issues that could be addressed by users employing different applications. Hong and Landay (2004) created a toolkit for developers called "Confab" that facilitates the development of privacy-sensitive ubiquitous applications. Confab includes a framework and customizable privacy mechanisms. The tool also includes extensions for managing location privacy. The goal of Confab is to enable application developers to address the range of trust and privacy requirements of end users of ubiquitous computing systems.

There is a growing area of research related to HCI issues and privacy in electronic voting. Hochheiser, Bederson, Johnson, C. Karat, and Lazar (2005) analyzed the current electronic voting systems and determined that increased usability was needed in these systems in three areas. First, voters must have information available to them about the electronic voting machines prior to arriving at the polling place and while there. Second, voters must be able to easily and confidently understand the machine, read the contents of the ballot, and cast their votes. Finally, voters must be able to trust the fairness and accuracy of the electronic voting system. A critical factor in trusting the system is being able to verify their votes without sacrificing their anonymity. Thus, privacy is a critical requirement in the design of successful electronic voting systems. Voter-Verified Paper Trails (VVPTs) can provide auditability of individual votes. It is also crucial to have well-designed audit trails regarding large-scale aspects of the system such as configuration, installation, and voter-count. Shubina and Smith (2004) conducted research on how to design coercion-resistant, voter verifiable electronic voting systems. They stated that it is important that voters can verify that the tally reflects the sum of votes actually cast and that voters can maintain the privacy of their votes and thus not be at risk of coercion from adversaries. They designed and prototyped a system to accomplish these goals. Their design ensured that voters are not subject to coercion from adversaries by providing a voter verification receipt that provides no real information without access to the voting machine or the contents of the cast ballots.

In terms of providing solutions for users to manage their privacy in the future, the European Union is sponsoring research on a system called PRIME (Privacy and Identity Management for Europe). The goal of the PRIME system is to support the user in controlling what PI about him- or herself is revealed to others through interactions with individuals, systems, and organizations (Pettersson et al., 2005). Pettersson et al. (2005) described three alternative user interface paradigms for privacy-enhanced identity management and illustrated how key legal privacy principles from the European Union Directives are associated with these designs. In the report of the first year of research in this effort, the team reported some results of initial usability evaluations of mock-ups of the three paradigms.

Organizational requirements in managing privacy. HCI research that is underway and focused on the organizational view of privacy includes the SPARCLE Policy Workbench (Karat, J., Karat, C., Brodie, & Feng, 2005) and efforts by Adams and Blanford (2005) in the healthcare domain. J. Karat, C. Karat, Brodie, and Feng employed user-centered design methods with 109 target users to identify organizational privacy requirements, and then they designed and tested a prototype system for users to author privacy policies, implement the policies, and conduct compliance audits of them. The prototype was a Wizard of Oz version of a system, meaning that users were able to have an immersive and dynamic experience with the system capability that seemed real, although there was no functional code behind the screens. Empirical results show that organizational users value highly a set of capabilities that enables policy officers to use natural language to author the policies, perhaps beginning with a template. Users found a visualization of the policies very valuable for communication, review, and reaching consensus across the organization. The implementation capability enabled the experts to approve the nominated mappings between rule elements and database fields and applications in the organization's configuration. The compliance audit capability enabled the users to run general audits to verify that the policy complied with regulations, to verify that the policy was enforced operationally as stated, and to run specific inquiries based on individual requests for information about use of personal information by the organization. The organizational users who were participants in the evaluation study rated the prototyped functionality as being of very high value to them.

C. Karat, J. Karat, Brodie, and Feng (2006) also conducted an empirical study to determine whether the two methods of authoring rules that were prototyped by J. Karat, C. Karat, Brodie, and Feng (2005; natural language with a guide or structured list) enabled users to create higher quality rules than an unguided natural language control method (using a word-processing window). Empirical results demonstrated that both prototyped authoring methods yielded higher quality rules than the control condition. With no training in use of the methods, users were able to author privacy rules, covering about 80% of the required rule elements described in scenarios when using either the natural language with a privacy rule guide tool or a structured list tool, as compared to covering about 40% of the required elements in scenarios using the unguided natural language control condition.

Research and development on the SPARCLE (Server Privacy Architecture and Capability Enablement) policy workbench is continuing. The general rule authoring utility for the privacy domain is functional, working code that has been successfully tested with the privacy policies of banking and finance, healthcare, and government organizations (Brodie, C. Karat, & J. Karat, 2006). Users can create new policies with SPARCLE, can import existing text versions of policies, or can cut and paste sections of policies to form a new policy. Then, SPARCLE employs natural language processing technology to parse and identify the rule elements. Users review and modify the rules, and when they are happy with the policy, the policy is transformed into XACML code (the OASIS international standard for the format of security-access control rules with a privacy profile) for input to the enforcement engine. The team is continuing to design and create other components in the end-to-end solution for privacy and believes that the policy workbench can be generalized to the security domain and possibly others.

Adams and Blanford (2005) addressed the gap between organizational and user perspectives of security and privacy in the healthcare domain. They conducted ethnographic evaluations of 93 clinical staff, managers, library staff, and IT department members in two hospitals. They found contrasting perspectives on security and privacy within the groups. They employed the concept of communities of practice to identify security and privacy issues within organizations, acknowledging the importance of knowledge gained through a work community in day-to-day work practices as well as formal learning. The community of practice acted as a link between the individual and the organization. In their research, one hospital improved communication and awareness across the organization through a community and user-centered approach to the design and development of an organizational privacy and security application. In the second hospital, a clash between the formal rules and the perspective of the local community of practice were at the heart of an identified security problem. The authors highlighted the importance of designing security and privacy practices within the context of communities of practice.

Summary of Findings and Recommendations

There is a range of HCI research that can inform the usability of the design of privacy mechanisms (see Table 32.1). The review has covered topics related to the networked world in which we live, current privacy legislation, individual differences in defining the meaning of privacy, end user's concern with anonymity while interacting over networks and in terms of the risks of aggregation of data, end user desire for personalization, the impact of some technologies (e.g., pervasive devices, P2P, P3P) on privacy, and end user and organizational views of privacy in a variety of domains. The central theme that emerges is informed user control over the disclosure of personal information. As presented in this section, different views and situational contexts regarding privacy ensure that this is a multifaceted challenge for experts in the HCI and privacy fields to address. In our roles as usability professionals, it is incumbent on us to consider the issues of privacy in the design, review, approval, development, and use of computing systems with which we are involved.

The HCI field will be enriched by considering privacy more fully, and this research will cross-pollinate in several areas. For example, cross-cultural studies of privacy inform the growing interest in HCI in cross-cultural interfaces and coordination. Kumaraguru and Cranor (2005) conducted a study about privacy in India, and the results demonstrated an overall lack of awareness and concern about privacy issues among users as compared to studies of users conducted in the United States. Increased understanding of the diversity and complexity of user preferences, and potential clusters of variables of interest, is providing new impetus for HCI research on individual differences. Moreover, considerations of visualizations and intelligent tutoring systems for privacy in a range of applications and tasks could germinate new emphases in HCI. Privacy in information

TABLE 32.1. Design Factors Impacting Usable Privacy Solutions.

Variable	Impact on HCI and Privacy
Networked World	• There is technology and data collection in most aspects of everyday life in the developed world.
Privacy Legislation	• Requirements for privacy vary by geography and across cultures. Organizations and end users must become knowledgeable.
Individual Differences	• People perceive and define privacy intrusions differently, and privacy solutions must be designed to support varying definitions and choices.
End User Desire for Anonymity	• End users must be informed of the risks of choices involved in different anonymity-enabling situations.
End User Desire for Personalization	• User control of data to be disclosed is critical for using personalization.
Pervasive Devices/Ubiquitous Computing	• Users want control over access to them but want it to be easy to use, fast, and flexible.
Platform for Privacy Preferences (P3P)	• Can inform user of any issues between their own policy and policy of a website they are considering interacting with. • No verification of adherence to policy possible at this time.
Peer-to-Peer (P2P)	• User must be clearly informed about data at risk of disclosure.
Electronic Voting	• User must understand how anonimity is protected and verification is enabled.
Organizational Management of Privacy—Requirements, Approaches, Perspective	• Authors of privacy policies, who have the knowledge of organizational practices, must be able to tie written policies to implementation through technology, with verification through compliance audits. • Critical to understand community of practice issues in an organization in creating privacy polices.

technology is likely to remain a critical issue for the near future. As it directly engages aspects of user control and power, privacy is central to the concerns of HCI. HCI and privacy will to remain closely interlocked, and in their intersection are the seeds for successful solutions.

Views on Security

It is broadly recognized that one of the major challenges to the effective deployment of information security systems is getting people to use them correctly (CRA, 2003). As far back as the 1970s, usability with specific reference to security mechanisms was identified as a key principle in the design of secure systems (Saltzer & Schroeder, 1975). Even beyond the domain of electronic information systems, there are many examples of the fact that overly complex security systems actually reduce effective security. For example, Kahn (1967), cited by Anderson (1993), suggested that Russian military disasters of World War II were partly because Russian soldiers abandoned the official army cipher systems because they were too hard to use, and instead reverted to simpler systems that proved easier to crack. Schneiner (2000) summed up the situation: "Security measures that aren't understood by and agreed to by everyone don't work" (p. 373). However, as with many areas in the design of complex systems, recognizing a potential problem does not necessarily create a rush of work to resolve it. Work on making security usable—and balancing the complex relationship between secure and easy to use—is just beginning to become a major topic.

Networked computer systems are increasingly the site of people's work and activity. Millions of ordinary citizens conduct commercial transactions over the Internet, manage their finances, and pay their bills online. Companies increasingly use the Internet to connect different offices and to form virtual teams to tackle mission-critical problems through virtual interaction. For example, interaction between citizens and local and federal government agencies can increasingly be conducted electronically; and the 2004 national elections in Brazil and (to a much more limited extent) the United States saw the introduction of electronic voting, which will no doubt become more widespread.

However, these new opportunities have costs associated with them. Commercial, political, and financial transactions involve disclosing sensitive information. The media regularly carries stories about credit card fraud, identity theft, and hackers breaking into commercial servers. Many people are nervous about committing PI to electronic information infrastructures. Even though modern PCs are powerful enough to offer strong cryptographic guarantees and high levels of security, these concerns remain.

The need for secure systems is broadly recognized, but most discussions of the problem of security focus on the foundational elements of information systems (such as network transmission and information storage) and the mechanisms available to system developers, integrators, and managers to ensure secure operation and management of data. Security, though, is a broader concern and a problem for the end users of information systems as much as for their administrators. Participation in activities such as e-commerce requires that people be able to trust the infrastructures that will deliver these services to them.

This is not quite the same as saying that we need more secure infrastructures. De Paula et al. (2005) suggested that it is important to separate theoretical security (the level of secure communication and computation that is technically feasible) from effective security (the level of security that can practically be achieved in everyday settings). Levels of effective security are usually lower than those of theoretical security. A number of reasons for this disparity have been identified, including poor implementations of key security algorithms (Kelsey et al., 1998), insecure programming techniques (Wagner et al., 2000; Shankar

et al., 2001), insecure protocol design (Kemmerer et al., 1994; Schneier & Mudge, 1998), and inadequate operating systems support (Ames et al., 1983; Bernaschi et al., 2000).

One important source of the disparity, though, is problems around the extent to which users can comprehend and make effective use of security mechanisms. Approaches that attempt to make the provision of system security automatic or transparent essentially remove security from the domain of the end user. However, in situations where only the end user can determine the appropriate use of information or the necessary levels of security, then this explicit disempowerment becomes problematic.

Perhaps the best consolidation of usability issues for security comes from Sasse and Flechais (2005). They observed that currently users often disclose (or write down) passwords, fail to encrypt confidential messages, switch virus checkers off, and generally engage in behavior counter to good security, and they asked why this is so common. They concluded that most users:

1. Have problems using security tools correctly
2. Do not understand the importance of data, software and systems for their organizations
3. Do not believe that the assets are at risk
4. Do not understand that their behavior puts assets at risk

Whitten and Tygar (1999) identified a "weakest link property" stating that attackers need only to exploit a single weak point in system security. Frequently, the human user proves to be this weakest link—not from malicious intention but from inability to reasonably do the right thing. Sometimes this is a mater of education—people need to be made aware of what to do and why. Some of the blame and burden on making our systems more secure rests in making security systems more usable.

Usability of Security Software and Mechanisms

In a series of studies, researchers at University College, London have explored some of the interactions between usability and security (Adams, Sasse, & Lunt, 1997; Adams & Sasse, 1999). They focused on user-visible elements of security systems, such as passwords. Although many information systems professionals regard users as being uninterested in the security of their systems (and, indeed, likely to circumvent it by choosing poor passwords, etc.), Adams and Sasse's investigations demonstrated that users are certainly motivated to support the security of the system but often are unable to determine the security implications of their actions. The specific problems that they identify with passwords have also led to interesting design alternatives (Brostoff & Sasse, 2000; Dhamija & Perrig, 2000).

In some cases, the complexity of making security work is as much a matter of interface design as anything else. Whitten and Tygar (1999) presented a usability analysis of PGP 5.0, demonstrating the difficulties that users have in completing experimental tasks (in their user study, only 3 out of 12 test subjects successfully completed a standard set of tasks using PGP to encrypt and decrypt e-mail.) The problems that they uncovered were largely problems of interface design, and in particular, the poor matching between user needs and the structure of the encryption technology provided to meet these needs.

Zurko and Simon (1996) explored similar concerns in their focus on user-centered security. Their work addressed their perceptions that the inscrutability of conventional security mechanisms makes it less likely that users will employ them effectively. The approach they outlined focused on graphical interfaces and query mechanisms to MAP, an authorization engine. While this approach is clearly helpful, it is limited to a particular area of system security and lacks the real-time feedback.

Control Over Security

One area at the intersection of usability and security that has received some attention is the role of access control in interactive and collaborative systems. For example, Dewan and Shen (1998; Shen & Dewan, 1992) explored the use of access control and meta-access control models as a basis for describing and controlling degrees of information access and management in collaborative systems. This is not simply a technical matter since the structure and behavior of these internal components can have a significant effect on the forms of interactivity and collaboration they can support (Greenberg & Marwood, 1994).

Many collaborative systems involve privacy issues and need to provide users with control over the disclosure of information. This has spurred a number of researchers to explore the development of privacy control systems that are tailored to the needs of end users. For instance, Dourish (1993) described the relationship between three different security mechanisms for similar multimedia communication systems, each of which reflects assumptions and requirements of the different organizations in which they were developed. Bellotti and Sellen (1993) drew on experiences with multimedia and ubiquitous computing environments to identify the source of a number of potential privacy and security problems. Their primary concepts—disembodiment and dissociation—were visibility problems, related to the disconnection between actors and actions that rendered either actor invisible at the site of action, or actions invisible to the actor.

Based on their investigations of privacy problems in online transactions, Ackerman and colleagues (Ackerman, Cranor, & Reagle, 1999; Ackerman & Cranor, 1999) proposed the idea of privacy critics—semiautonomous agents that monitor online action and inform users about potential privacy threats and available countermeasures. Again, this mechanism turns on the ability to render invisible threats visible.

One important related topic is control over the degree of security available. One of our criticisms of traditional security systems has been their all or nothing approaches. However, some work has attempted to characterize degrees of security provision, as embodied by the idea of "quality of security service" (Irvine & Levin, 1999; Spyropoulou et al., 2000). This builds on earlier work, establishing taxonomy of security service levels (Irvine & Levin, 1999). The fundamental insight is that organizations and applications need to trade off different factors against each other, including security of various forms and degrees, in order to make effective use of available resources (Thomsen & Denz, 1997; Henning, 1999). While this work is directed towards resource management rather than user control, it begins to unpack

the security black box and characterizes degrees and qualities of security.

For end users, perceived security can be defined as the level of security that users feel while they are shopping on e-commerce websites. Yenisey, Ozok, and Salvendy (2005) reported a study that aimed to determine items that positively influence this feeling of security by users during shopping and to develop guidelines for perceived security in e-commerce. An experiment allowed users with different security assurances to shop on simulated e-commerce websites. The participants were divided into three groups—shopping for cheap, midrange, and expensive products, respectively. Following the shopping environment, a virtual shopping security questionnaire was presented to the users. Generally, there were no significant differences in item ratings between the groups of different shopping item values. A factor analysis procedure determined two main factors concerning perceived security in e-commerce. The perceived operational factor includes the website's blocking of unauthorized access; emphasis on login name and password authentication; funding and budget spent on security; monitoring of user compliance with security procedures; integration of state-of-the-art systems; distribution of security items within the website; website's encryption strategy; and consolidation with network security vendors. The perceived policy-related factor includes the website's emphasis on network security; top management commitment; effort to make users aware of security procedures; the website's keeping up-to-date with product standards; the website's emphasis on security in file transfers; and issues concerning the Web browser.

Key Topic Areas for HCI *in Security*

Secure design and administration. Yee (2005) provided guidance for designing and evaluating usable secure software aimed at protecting the interests of the legitimate user. Starting with the work of Saltzer and Schroeder (1975), he divided his guidelines into two general categories—guidelines for authorizing others to access valuable resources and guidelines for communication between the system and the user. For authorization, he suggested associating greater risk to the user with greater effort or less visible operations so that user's natural tendencies leads to safe operations. For communication, he suggested enabling users to express safe security policies that fit their tasks. The guidelines are not empirically derived, but are gleaned from the experiences of security software designers.

Kandogan and Haber (2005) applied ethnographic approaches to the study of a particular class of users involved in the protection of systems—security administrators. The issues are somewhat different for populations whose job responsibilities include the security of systems used by others when compared to the situation in which users are acting as their own security administrators. However, they represent another important category of user in the area of creating usable security systems. In the ethnographic tradition, they examined their users in typical work situations (e.g., detecting and addressing a security attack) and looked at how well their tools help them to do their job. One important result of their work was the realization that administrators simply have too much to look at in the course of their daily work and that they specifically need tools to help them understand vast amounts of dynamic data. Their work focused on developing visualization tools to assist in this situation.

Balfanz, Durfee, Smetters, and Grinter, (2004) presented guidelines for usable security systems developed over a course of research that has focused on the intersection of security and usability. The guidelines included advice to focus on designing both security and usability into the system rather than attempting to retrofit either into existing systems. It seems that the lessons learned in HCI work in regards to the need to consider it early and often during design also apply to security (security engineering is an important topic in that field, much as usability engineering is in HCI). Additional guidelines also called for a focus on the user by looking for high-level building blocks that can be used to create user-oriented solutions, rather than assuming users will be skilled at assembling their own tools for effective security. They pointed out that the security community has long held the belief that security is more important than user needs and that users must adapt to requirements in order to assure system security. An empirical study (Balfanz, Durfee, Grinter, Smetters, & Stewart, 2004) showed how hard it is for security researchers to accept the difficulty of using some of their own systems. When they evaluated the time it took to secure a wireless network using a tool they had developed, they were surprised to find that it took users over two hours on average. The authors mentioned that the reaction in the security community was often that the empirical data must be wrong. While these views are changing with the emergence of usable security research groups like this one, we should not expect the difficult trade-offs to be quickly resolved.

Maxion and Reeder (2005) provided a study in which a system specifically designed to assist users in avoiding errors (Salmon) was compared to the standard operating system interface for making file-sharing decisions. Salmon was found to increase successful task completion by 300% (the standard interface used misleading terminology, which caused users to feel that they were sharing files when in fact they were not). Users also spent less time searching for information using Salmon and had a greater proportion of their time on essential task steps rather than security overhead. In their study, they demonstrated that attending to error avoidance in interface design could facilitate usable security.

Authentication mechanisms. Security systems are designed to let authorized people in (the permission problem), and to keep unauthorized people out (the prevention problem). This involves three distinct steps: identification, authentication, and authorization (Renaud, 2005). The identification step asks a person to identify himself—usually by means of a token or an identification string such as an e-mail address or account number. Once the identification token has been provided, the person has to provide some evidence of his or her identity (authentication). This can be done by presenting something he or she knows (e.g., password), something he or she recognizes (e.g., graphical passwords), something he or she holds (e.g., a certificate), or something he or she is (e.g., biometrics). For the first three, authentication depends on the user and the system sharing a secret, which for security purposes should be difficult for someone else to guess, and for usability

purposes should be easy for the user to remember. The trade-off between these two purposes can be difficult to resolve, and contributes to discussions of how much security rather than a view of security as something that is either present or not. In the case of biometrics, the system records a digital representation of some aspect of a person's physiology or behavior at enrollment, and this is confirmed at authentication time.

Many authentication scenarios can be strengthened (in a security sense) with public key cryptography. For example, a user can have a smart card that contains a public key and a matching private key. Instead of a password, the user's public key can be placed on file at a remote computer system (authentication server). To authenticate the user, the remote system sends the user a random challenge. The user signs the challenge with his private key and sends the result back to the remote server, which verifies the signature with the public key. In this way, the remote server can verify that the user has possession of the private key without having to receive it. Instead of having the public key on file at the remote system, the smart card can submit both the signed challenge and a public key certificate that has been signed by a third party. In this case, the use of public key technology is called public key infrastructure (PKI).

Whatever methods are used, at each stage of an authentication process, we can ask, "Is it secure?" The real areas of vulnerability are the input mechanism and the user. In the case of knowledge-based authentication, the user must be able to keep the secret and the secret must be hard to discover.

Passwords. Random passwords are currently the most popular user authentication mechanism. As such, they also represent the most common research target for work in the security and usability arena. One might consider them the white rat for HCI work, similar to menu structures or word processor designs of earlier HCI eras. Yan, Anderson, Blackwell, and Grant, (2005) conducted research to examine some of the commonly held beliefs about the security and usability properties of various password generation guidelines. In their work, they confirmed that passwords generated by typical advice (e.g., use of a random sequence of letters and numbers) can be difficult for people to remember, but that more cognitively friendly advice (such as select mnemonic-based passwords) can be much easier to remember without sacrificing the theoretical security levels of random passwords. They also explored additional security screening capabilities—such as screening for weak choices (e.g., obvious dates/places)—and found that such filtering could work well with other advice provided to users. Contrary to beliefs held within security communities, they did not find that theoretical analysis necessarily corresponded to actual system security. For example, encouraging the use of truly random passwords resulted in more frequent writing down and carrying passwords than encouraging the use of mnemonic passwords.

Others have taken up the challenge of providing usable passwords through nontext password schemes. For example, Monrose and Reiter (2005) provided an analysis of graphical passwords. As with text passwords, graphical passwords might be selected by the user (possibly enhancing usability but perhaps decreasing security) or selected by the system (enhancing security at the cost of usability). While there is certainly ample evidence that people exhibit powerful memory for images (e.g., Mandler, 1991), it is not completely clear that this translates easily into a superior password mechanism. Results from Monrose and Reiter demonstrated that graphical password schemes suffer drawbacks similar to those of textual schemes. People tended to select memorable graphical passwords enabling them to be more easily attacked, and random graphical figures can be more secure but also more difficult to recall.

Wiedenbeck, Waters, Birget, Brodskiy, and Memon, (2005) experimented with an interesting variant of graphical password. In their PassPoint system, users are provided with an image in which they establish a password by selecting a series of points on the image (the number of points selected and the order in which they are selected are factors in the complexity—and thus the potential security features of the password). Their experimental work indicated that such a system is promising from a usability perspective—people can learn the systems easily and remember their sequences over time. However, performance with such a system was not as good as with a comparison textual password system. It took more time to learn the passwords, and more time to enter them even with practice. The extent to which such performances differences might be due to novelty associated with graphical systems compared to textual systems remains unclear, and is in need of further research before we might expect them to replace the textual passwords with which we have so much experience. As De Angeli, Conventry, Johnson, and Renaud (2005) reported in their work on the graphical authentication systems, successful design of password systems is a complex task and requires considering and weighting a number of factors (such as the trade-off between security and usability).

Some of the trade-offs between security and usability, and some of the tension between the approaches of the two communities can be seen in some recent research lines. Davis, Monrose, and Reiter (2004) provided a study that looked at how user selection of passwords in graphical password schemes might increase the likelihood that passwords can be attacked. For text passwords, user selection is the norm, and there is no evidence that allowing users to select passwords (following some guidelines for length and character type) makes the resulting passwords any easier to brute force guess than the theoretical maximum set of passwords for the set of characters involved. Several new approaches for password selection—including a number that involves allowing the user to select a sequence of images—have been developed to help with password memorability (for a commercially available example, see Real User Corporation, 2002). While there has been research on several of these approaches to indicate that they provide passwords that are more memorable than text passwords, and have similar theoretical protection (entropy), there has not been much research into how effective they are in actual use. The research reported by Davis et al. suggested that people do not select as randomly, and instead form passwords with a limited set of possibilities (perhaps analogous to forming passwords with strings that might be easily associated with an individual such as birthdates). Overall, many questions remain regarding how to optimize both security and usability at the same time in the area of knowledge-based passwords.

Biometrics. As we have discussed, authentication in computer systems has suffered from the limited number of types of

mechanisms available for a system to know who it is interacting with. With advances in technology, we have new opportunities (beyond traditional mechanisms such as passwords) for systems to recognize us. Biometrics refers to a means of identification that can be uniquely associated with an individual (e.g., voice patterns, fingerprints, and hand geometry). Biometrics involves the comparison of live anatomical, physiological, or behavioral characteristics to a stored template of a person (Coventry, 2005). Physiological biometrics that have been investigated include those based on fingerprints, hand and/or finger geometry, and the patterns of retinas, veins, irises, and faces. Behavioral biometric techniques that have been advanced include those based on voice, signature, and typing behavior (Peacock, Ke, & Wilkerson, 2005). Other techniques, which are not yet as well developed, include recognition of characteristics such as ear shape, gait, laughter recognition, facial thermograms, and lip shapes.

To begin, it is important to distinguish between the use of biometrics for identification and verification. For identification, the task is to identify an individual out of a population of all possible users. As the population of possible users grows, the demands on identification can also grow, in terms of either performance or accuracy of the recognition system—challenges similar to speech recognition for large vocabulary unconstrained speech. For verification, the task is to verify a particular identity by matching a characteristic to a stored template for the individual. The computational task is much simpler for verification than for identification (where only fingerprints, retinal scanning, and iris scanning have been proven successful for large populations). For the most part, we will talk about the use of biometrics for verification in this chapter.

Coventry (2005) provided an excellent illustration of how biometrics might be used and what some of the current trade-offs are for a common application—automatic teller machines (ATMs). For all techniques, usability issues can be cited as barriers to implementation and acceptance. Fingerprint identification is relatively well developed, but sensors placed in public places are subject to environmental issues (such as dirt or latent prints) as well as user training issues (they can be sensitive to where the finger is placed on the reading device). Retina scans have attractive potential, but are currently quite invasive and difficult to develop for a full range of users (they require placing the eye close to a sensor location). Speaker verification can be subject to background noise problems and relatively complicated user registration requiring more input than a single word or phrase. Signature verification is attractive because of its long use and association with financial authorization, but current technology is not yet seen as reliable. Typing verification is seen as having some potential, but lacking the reliability necessary for large population verification.

Biometrics researchers have determined that real-life users are the biggest variable in system performance. Ashbourn (2000) suggested several user characteristics for evaluating biometric systems. These included a users general acceptance of the biometrics concept employed (e.g., Is the idea of having a system read such characteristics acceptable?), general knowledge of the technology (e.g., Does the user understand what is being done?), knowledge of the particular biometric characteristic (e.g., Does the user understand that, if fingerprints are being read, his or her finger should be centrally placed on the sensing device?), experience with the sensing device, environment of use (e.g., public or private), and transaction criticality.

While biometrics technologies are improving rapidly, inherent performance limitations remain and are extremely difficult to work around, except perhaps by combining multiple technologies or providing for a bypass.

Other security approaches. One other way of authentication by something I have is with smart cards—here referring to a portable device (or card) with processing power and authentication information contained on it. Smart cards essentially add an integrated circuit to familiar plastic credit cards enabling the use of cryptographic services like random number generation and public key cryptography (Piazzalunga, Salvaneschi, & Coffetti, 2005).

Just (2004) examined the design of challenge question systems, looking for ways to improve the usability/security characteristics of these. Beckles, Welch, and Basney (2005) looked at the usability of security mechanisms in grid computing contexts.

Encryption of data is a well-established mechanism for protecting information. Rather than storing or transmitting data which can be read in the clear (e.g., data which is stored in standard code schemes such as ASCII), information is encrypted using a key and then must be decrypted using a key. While this contributes to achieving some security goals (for example, if an encrypted file containing sensitive information is lost or stolen, it would be a considerably harder task to make use of the information than if an plain data file were lost or stolen), there is a usability cost for this added security—users might have to indicate when and how files are to be encrypted or decrypted. Unfortunately, the complexity of this process has been found to cause user difficulties, variously resulting in inability to use security mechanisms appropriately (Whitten & Tygar, 1999; Caloyannides, 2004; Guttmann, 2003). Work continues addressing the issues associated with cryptographic approaches to security (e.g., Balfanz, Durfee, & Smetters, 2005).

Usability Challenges Specific to Security—Attack Defense

Much of what we present above should seem familiar to the usability practitioner in the sense that security considerations should be considered as part of the overall system use context. The choice of authentication technique in general should consider the various trade-offs in the techniques available. Biometric techniques can be considered if users are likely to accept them and if they cover the population of expected users. If passwords are employed, they should be easy for users to recall and enter but difficult to guess or steal. Some specific issues arise because of the nature of security as a protection against attack. The fact that it is not just the user, but also that other parties with intention to do harm that must be considered, makes considering usability issues complicated in new ways.

For the majority of usability work, the goal is to make the user's primary task easy. For security considerations, we encounter a different situation—the fact that people might be trying to deceive the user and commit fraud (Conti, Ahamad, & Stasko, 2005). Of considerable recent interest in the security arena is the battle against phishing—the misrepresentation by someone of an

identity or website intended to draw the user into interactions that can be harmful to the user. A phishing attack succeeds when a user is tricked into forming an inaccurate model of the interaction and when the user takes actions contrary to their intentions. Attacks often begin with e-mails sent to potential victims, purporting to be from an individual or organization with whom that the user has some legitimate reason for interacting.

The issue of phishing has been viewed as a model problem for illustrating usability concerns in security (Dhamija & Tygar, 2005; Miller & Wu, 2005). An Anti-Phishing Working Group (2005) collected and described phishing attacks. Analysis of the reasons why phishing attacks are so often successful points to a number of interesting usability aspects. For example, operating systems and windowing platforms that permit general-purpose graphics also allow attackers to mimic the appearance of legitimate websites. Users tend to habituate on commonly occurring warnings about submitting data over unencrypted connections and can easily fail to notice when they are actually entering information to an insecure website. Because organizations can invest heavily in having their names and logos associated with trust, attackers can take advantage of this association by simply convincing the user that they represent a trusted organization. Because security is generally a secondary goal for users, there is only so much attention we can expect them to pay to attempts at fraud.

End users are not alone in trying to prevent attacks on systems. For people working in organizations with system administrators, the usability problem is partly moved from the end user to the administrator. Here the focus shifts to making tools for users whose primary activities include monitoring the health of the overall system. Security is one of their main tasks. Intrusion detection—the problem of detecting computer attacks in a timely manner—is one of both great difficulty and utmost importance. Finding specific evidence of attack activity in the enormous number of potentially relevant alerts, packets, operating system events, and user actions presents an almost overwhelming task for intrusion detection (ID) analysts. Visualization tools and techniques can be used to increase the effectiveness of intrusion detection analysts by more fully exploiting their visual cognition abilities. Recent work identified promising avenues for research in visualization for intrusion detection (e.g., Goodall, Ozok, Lutters, Rheingens, & Komlodi, 2005). Goodall et al. presented a user-centered visualization based understanding of the work of ID and the needs of analysts derived from the first significant user study of ID. Consistent with good information visualization practice, their tools presented analysts with both at-a-glance understanding of network activity and low-level network link details. Results from preliminary usability testing show that users performed better and found easier those tasks dealing with network state in comparison to network link tasks.

Summary of Findings and Recommendations

We would all like our information to be secure. That is why we would like our systems to be resistant to access or modification from individuals who we do not authorize to have access to them. This requires that users be aware of security mechanisms even though they would generally prefer not to. The design challenge is to make security appropriately usable—that is to make it as easy to use as possible for the intended users, while making it as difficult as possible to circumvent for the potential attacker. This means that security administration and user involves trade-offs that need to be evaluated in the design of a system. Absolute security is a myth—we need to understand that we are looking for appropriate levels of security.

Some of the aspects of security that researchers have investigated are mentioned above and summarized below in Table 32.2. We think that serious consideration of the trade-offs between usability and security just began and that much work remains. While initial approaches involved hiding security controls from users, we believe that this needs to be done in balance with giving the user the ability to control the level of security required.

Views on Trust

Trust is important to every relationship that exists in society, both those of personal and public nature. Trust is related to security and privacy but differs from these concepts in that it is not an objective measure. Trust is based on the perception that the person, organization, or system one is dealing with is reliable. There are many definitions of trust and none of them satisfies every use. These definitions vary across academic domains and with context. When dealing with e-commerce and other computer applications, the user is deciding not only to trust the individual or organization, but also the technology implementation provided by the individual or organization. Kuhlen (1998) defined trust as "allowing us to act as if we have

TABLE 32.2. Design Factors Impacting Usable Security

Automatic Security Versus User Controlled Security	• Security is not generally the main user task focus for PIM. • Risk Management is not a topic that end users explicitly understand.
Authentication Mechanisms	• Access to resources is controlled by knowledge of who the user is. • Identification relies on authentication through one (or more) mechanisms. • Mechanisms have usability/security trade-offs. • Can be "something I know" or "something I have."
Passwords	• Usually textual, but can be other. • Users will forget passwords. • Users will select passwords that are not optimally secure.
Biometrics	• A form of "something I have" authentication. • Not all users might have a characteristic (e.g., fingerprint). • Recognition technology is involved, so errors and attacks are possible.
Attack Considerations	• Complete design requires consideration of possible attacks. • Anti-phishing approaches • Visualization for system administrator tools

perfect knowledge about the reliability of that entity even if we do not" (p. 3). Marsh and Dibben (2003) defined trust as concerning "a positive expectation regarding the behavior of someone or something in a situation that entails risk to the trusting party" (p. 466). Jøsang and Presti (2004) defined trust as "the extent to which one party is willing to depend on somebody or something, in a given situation with a feeling of relative security, even though negative consequences are possible" (p. 1). All of these definitions share the concept that trust is about allowing us to feel comfortable and allowing another entity, such as an organization or an organization's computer system, to take one or more actions for us even though there is risk that we will be harmed in some way. In other words, in the domain of e-commerce, the trusting party believes that the trusted organization has implemented privacy and security so that it will protect the PI of the individuals interacting with it.

Earlier in this chapter, we discussed the ongoing research into security and privacy functionality that is intended to protect information that is collected and stored by organizations. Therefore, in this discussion, we will assume that privacy and security protections are in place, and we will concentrate on the factors that affect individuals' perception of trust.

Factors Influencing Trust

Many researchers have studied the factors that influence trust when dealing with online applications. Brodie, C-M. Karat, and J. Karat (2004) showed that greater degrees of transparency and control regarding the use of personal information do increase website visitor trust in the domain of IT equipment. In addition, to control of one's own data and transparency of its use, researchers have found that an individual's familiarity with a website and the perceived credibility and quality of the website also affect trust. Fogg et al. (2001) found that website visitors' perceptions of the credibility of websites were most influenced by the degree to which it was connected to a known organization in the real world and whether it seemed well designed and implemented. Building on this research, Fogg, Soohoo, Danielson, Marable, Stanford, and Tauber (2003) conducted a study with 2,500 participants in order to understand factors that influence the perceived credibility of a website. They found 18 factors that influenced credibility. The biggest factor by far was the design of the appearance of the website. This was followed by the information design or structure, information focus, perceived motivation of the website owner, perceived usefulness of the information, accuracy of the information, name recognition and reputation of the website, the tone of the writing, the identity of the sponsor, functionality on the website, customer service, past experience with the website, clarity of the information, performance on a test, readability, and affiliations (Fogg et al., 2003). They attributed the importance of the design look of the website to the fact that it is very prominent when a visitor looks at a website. This supports Fogg's (2003) theory that credibility is based on noticing a feature and then interpreting its quality to create judgment. This theory is posited as a reason why there are apparent differences in findings relating to the affect of privacy guarantees (e.g., privacy statements and privacy seals) on websites (Palmer, Bailey, & Faraj, 2000; Turow, 2003). Palmer et al. found that privacy seals and statements did contribute to user's trust of website, while Turow found that website visitors do not read privacy statements or understand the meaning of seals and that there is a need to educate the public regarding privacy issues and the web.

Given that familiarity is a factor in establishing trust, Zhang and Ghorbani (2004) identified several factors that influence familiarity, including the individual's knowledge of similar services (prior experience), the number of times he or she has visited the particular website (repeated exposure), the length of each visit to the website (level of processing), and the interval of time between visits (forgetting rate).

Corritore, Kracher, and Wiedenbeck (2003) created a model using similar factors as influencing trust in the online world. In this model, trust is determined by both external factors and the user's perception of the application or website. The external factors include the user's propensity to trust and prior experience in similar situations and the user's perceptions include his or her perception of the credibility of the website, how easy it is to use, and the degree of risk involved (Corritore et al., 2003). Briggs, Simpson, and De Angeli (2004) looked at trust in a narrower domain—advice websites. They developed a model of how people determine whether to trust the advice they receive from websites. The authors developed a 22-point scale designed to break down trust into a set of judgments that can be measured. This list includes whether the user perceives the information to be prepared by an expert or a knowledgeable source, whether comments from other users were available on the website, if the website was owned by a known and respected company, if they had to wait a long time on the website, if different options were suggested by the website, if the website was perceived as hard to use, whether the user felt involved in how the website constructed the advice offered, if the site was perceived to be interactive, if the advice was tailored to the user, if the reasoning was explained to the user, if there was an opportunity to contact a human, if the advice appeared to be impartial, if the advice was perceived to be good and the user trusted it, and if the website behaved in a predictable way (Briggs et al, 2004). This study supported an earlier study that found trust was influenced by the perceived credibility of the source of the information, whether the website was personalized so that information was tailored to the individual, and whether the website was operated in a predictable way (Briggs, Buford, De Angeli, & Lynch, 2002). While many have found the trust affects the use of personalization, Briggs, Simpson, and De Angeli (2004) found that the presence of personalization could contribute to trust if it allowed the visitor to feel that the website is tailored to their needs. Sillence, Briggs, Fishwick, and Harris (2004) had similar results when studying trust of information on health related websites. They proposed a three stage model of trust in which visitors make initial, rapid assessment of a website based on the design and appearance of the website and then do a more systematic evaluation in which the credibility of the website and the degree to which it appears to be personalized to their situations becomes more important. The participants were looking for information written by or for people in situations similar to their own. The third stage of this model addressed the

maintenance of the trust relationship over time and left future work in this paper (Sillence et al., 2004).

Patrick, Briggs, and Marsh (2005) provided a thorough overview of many issues related to trust, including credibility of information on a website, familiarity with the website, and external factors such as prior knowledge and disposition to trust. They also described how trust is developed slowly over time, but a bad experience or even a processing error can destroy that trust very quickly. They reviewed a number of privacy models that have been operationalized using questionnaires, including Bhattacherjee's (2002) model that described how familiarity leads to trust and to a willingness to do business with a website and Corritore et al.'s (2003) model that is described above. Patrick et al. (2005) concluded with design guidelines for promoting trust based on the literature that they reviewed.

Trust and Risk

Trust cannot be considered without considering its relationship to risk. As Patrick et al. (2005) pointed out, "Trust is intimately associated with risk" (p. 81), and people evaluate the degree of risk in many situations both online and offline. In the online world, people need different skills to judge the trustworthiness of websites and applications and the organizations that create them correctly. People can become victims of phishing attacks and identity theft if they trust too much. However, Friedman, Kahn, and Howe (2000) pointed out costs to trusting too little, as well as opportunities that can be missed if users trust too little. Friedman et al. discussed several issues that make trust and risk decisions more difficult than face-to-face transactions such as the anonymity of online transactions, the difficulty in assessing the reliability and security of technology, and misleading statements and images. Patrick et al. (2005) referenced work by Chaiken (1980) that described two strategies used depending on the perceived level of risk involved. Where people are not highly involved with the decision, they often make decisions based on appearances, and when they are more deeply involved, they employ strategies that are more systematic (Chaiken, 1980). This suggests that the degree to which users believe they are at risk will determine the strategies they use to assess the credibility of a website or system and the organization behind it. This model is similar to the three-stage model proposed by Sillence et al. (2004) in that, in the first stage of their model, users make quick decisions based on appearance, which is similar to Chaiken's description of when people are not highly involved. In the second stage, Sillence et al. described the use perceptions of the credibility of the information, which is similar to the systematic strategies that Chaiken described when people are more involved.

Trust and Personalization

In the last few years, many e-commerce companies have invested in personalizing the e-commerce websites in order to encourage business. However, personalization is only useful if users are willing to share data with the website, and this has not always been the case. This requires that users trust that owners of the website will not misuse their data. Hagan (2000) found that website visitors often show their distrust of internet websites by disabling cookies on their computers and entering incorrect information into online forms. Many researchers have studied what aspects of an application or website affect user's willingness to use personalized features of websites. The Office of the Federal Privacy Commissioner of Australia (2000) conducted a large study of the factors that affect whether individuals are willing to share information with a business or other organization. They found that it is important to individuals that they understand how their data will be used and that they have control over who has access to it. Welty and Becerra-Fernandez (2001) examined how information technologies can reduce transaction costs and increase trust by increasing the symmetry of information available to both sides of the transaction in a business-to-business (B2B) setting. Likewise, Brodie, C.M. Karat, and J. Karat (2004) showed that greater degrees of transparency and control regarding the use of PI do increase willingness to share data in order to get access to personalized features of websites in the IT domain. Overall, trust and personalization have a complex relationship in which the presence of personalization functionality that can provide website visitors with information tailored to their situations and needs has been found to lead to a greater degree of trust with a website. At the same time, website visitors must trust the website and the organization behind it enough to share personal information if the functionality is going to be effective. Research has shown that website visitors are willing to share more personal information when they understand how and by whom their data will be used and when they have the ability to control how it is used, which suggests that these are important factors for designers to keep in mind when designing their websites.

Summary of Findings and Recommendations

Trust is necessary in an uncertain world. Given today's world in which the Internet is playing a larger role all of the time while the risks from phishing scams, spyware, and viruses continue to grow, helping users to develop an appropriate level of trust is an important goal. While there has been a great deal of research on why and when people trust in the online world, there is more to do. HCI researchers and practitioners can help by continuing to build an understanding of what leads individuals to trust a website or an application. Table 32.3 contains a summary of the factors identified in the research cited in this chapter that have been found to influence trust in the online world to date. HCI practitioners and website designers need to understand the factors that influence trust and use this knowledge as they design websites and applications. Research to date has tended to focus on a few domains, such as e-commerce and online advice websites, and it is not clear to what extent the findings hold across cultural boundaries. HCI researchers can address these issues by determining to what degree the factors that influence trust generalize across domains and cultures.

TABLE 33.3. Design Factors Impacting Trust.

Perceived Credibility of the Website or Application	• Credibility has been found to be influenced by many factors. • The most common factors sited was website and information design, the perceive usefulness of the data, and reputation of the provider
User Familiarity with a	• Familiarity has been found to be influenced by prior experiences with similar sites, repeated exposure to the site, the level of involvement with the site as measured by time spend on each page, and the time between visits to the site.
Predictability of the Behavior of the Website	• Users tended to trust sites that behaved in a predictable fashion.
Perceived Quality of Information Provided	• Judgments of the quality of online advice is related to the perceived credibility of the site.
External Factors	• These include propensity of the user to trust and users' prior experiences in similar situations.
Ease of Use	• Users trusted easy to use sites.
Perceived Deree of Risk in Dealing with the Site	• Users were less likely to trust sites where they perceived more was at risk.
Site Personalized for the User	• Does it appear to be tailored for the individual or be created by a person in a similar situation to the individual?
Ability to Control Informattion Shared with Site	• Users were more willing to trust sites with personal information if they felt they had a control of that information.
Degree of Transparency Regarding Use of Shared Information	• Users were more willing to trust sites when they understood how there data would be used and by whom.
Presence of Privacy/Security Seals and Statements	• Users trust the simple presence of these seals, though it has been shown that these statements and symbols are often not read or understood.

CONCLUSIONS AND DIRECTIONS

The intersection of HCI, privacy, security, and trust is emerging as a critical area for research amid the backdrop of recent world events. Information is being improperly disclosed in ways that causes real harm to people through identity theft. Organizations are concerned about theft of proprietary data and documents. Citizens are becoming increasingly concerned about the use of the vast amounts of data that organizations and governments have about them. The rapid advancement of the use of information technology in industry, government, and academia makes it much easier to collect, transfer, and store sensitive information of a variety of types around the world. We are faced with technical challenges that result from inadequate consideration of security and privacy issues in designing our information systems, which if not addressed will result in a significant deterioration of the trust that people have in IT. While there certainly is considerable research in this area, particularly on the security side, it is becoming increasingly clear that really making our systems secure and enabling appropriate attention to privacy issues will require more than just a technology focus. While usability has been identified as a major challenge to moving the results of security and privacy research into use in real systems (CRA, 2003), it is not clear that this challenge has reached the interdisciplinary researchers needed to carry out the complex agenda of work.

We see privacy as a complex social issue concerned with individuals' rights to know what information is collected about them and how it might be used. It needs to be concerned with more than just the protection of PII, such as credit card numbers, national identification numbers, and other such identifiers. In isolation, one data element might not identify the individual to whom it relates, however, data aggregation techniques are changing the playing field. Research by Sweeney (2001; Malin & Sweeney, 2004) demonstrated that minimal amounts of information believed to be anonymous can be used to personally identify someone. Questions regarding who has what rights to information about us for what purposes become more important as we move toward a world in which it is technically possible to know just about anything about just about anyone. As Adams and Sasse (2001) stated, "Most invasions of privacy are not intentional but due to designers' inability to anticipate how this data could be used, by whom, and how this might affect users" (p. 50).

Privacy of information held in information systems depends on security—the ability to protect a system's resources from harmful attack or unauthorized access. Security is an essential component in systems that house personal information on employees, clients, constituents, students, customers, and patients. The role of security goes well beyond the protection of PI. For example, it includes corporate and national security information about trade secrets, business practices, payrolls, research, and development; strategy and logistics plans; fuel preserves; and similar things that could wreak havoc if placed in the wrong hands (e.g., by industrial competitors who wish to obtain unfair advantage; terrorists who wish to bring harm to business or government entities, bring down power grids, or destabilize banking and financial systems around the world).

The security functionality in these systems must be usable for the people in the range of user roles with different perspectives and skills who interact with it. If the security aspects of a system are so complex that individuals within the various user roles (e.g., technical users, business users, and end users) cannot understand it, then costly errors will occur. In the HCI field, there is a saying that if the user cannot understand the functionality, it does not exist. In security, the problem may even be worse than the functionality not existing because the organization may think data is secured when it is not. Deciding how we can design usable and effective privacy and security technology capabilities for the future includes philosophical, legal, cultural, business, social policy, human performance, and practical dimensions. Since at its core the goals of HCI and user-centered design are to understand the user, the user's tasks or goals, and the social and physical context in which the user is interacting with the system, all of these dimensions can be considered as within the scope of creating usable and effective privacy and security solutions.

HCI research in privacy and security is a critical area of focus in today's world. Communication and collaboration is occurring between people who have expertise in one or more of the areas of privacy, security, and HCI. This emerging community is beginning to work together to incorporate HCI in the design and develop-

ment process for usable and effective privacy and security solutions. Some unique challenges to HCI research are related to privacy and security as compared to other domains. First, the use of security and privacy solutions is generally not the user's main goal. Users value and want security and privacy functionality, but they regard this functionality as secondary to completing their primary tasks (e.g., completing a transaction or process). Second, as mentioned above, today's solutions need to be usable for different groups of users with differing skill sets as compared to the past when security solutions in particular were designed with a highly trained and dedicated technical user in mind. Third, the risk of the negative impact of usability problems is higher for security and privacy applications than for many other types of systems. Finally, users must be able to update security and privacy solutions easily to accommodate frequent changes in legislation, regulation, and organizational requirements.

References

Ackerman, M., and Cranor, L. (1999). Privacy critics: Safe-guarding users' personal data. In *Proceedings of the Conference on Human Factors in Computing Systems* (CHI 1999), 258–259. New York, NY: ACM Press.

Ackerman, M., Cranor, L., & Reagle, J. (1999). Privacy in e-commerce: Examining user scenarios and privacy preferences. In *Proceedings of the ACM Conference on Electronic Commerce,* 1–8, New York, NY: ACM Press.

Ackerman, M., & Mainwaring, S. (2005). Privacy issues in human-computer interaction. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 381–400). Sebastopol, CA: O'Reilly.

Acquisti, A. (2004). Privacy in electronic commerce and the economics of immediate gratification. *In Proceedings of the ACM Electronic Commerce Conference* (pp. 21–29). New York: ACM Press.

Adams, A., & Blanford, A. (2005). Bridging the gap between organizational and user perspectives of security in the clinical domain. *International Journal of Human Computer Studies, 63*, (1/2), 175–202.

Adams, A., & Sasse, A. (1999). Users are not the enemy: Why users compromise security mechanisms and how to take remedial measures. *Communications of the ACM, 42*(12), 40–46.

Adams, A., & Sasse, A. (2001). Privacy in multimedia communications: protecting uers, not just data. In A. Blandford, J. Vanderdonkt & P. Gray (Eds.), People and computer XV—interaction without frontiers. *Joint Proceedings of HCI2001 and ICM2001*, Springer, Lille, 49–64.

Adams, A., Sasse, M. A., & Lunt, P. (1997). Making passwords secure and usable. Proceedings of HCI on People and Computers XII. Springer, Berlin, 1–19.

Altman, I. (1975). *The environment and social behavior, privacy, personal space, territory and crowding*. Monterey, CA: Brooks/Cole Pub. Co., Inc.

Ames, S., Gasser, M., & Schell, R. (1983). Security kernel design and implementation: An introduction. *IEEE Computer, 16*(7) 14–22.

Anderson, R. (1994). Why cryptosystems fail. *Communications of the ACM, 37*(11), 32–40.

Anti-Phishing Working Group. (2005). *APWG Phishing Archive*. Retrieved 2006, from http://anti-phishing.org/phishing_archive.htm.

Ashbourn, J. (2000). *Biometrics: advanced identity verification*, London: Springer Verlag.

Associated Press. (2003, September 25). Pentagon spy office to close. *Wired News*. Retrieved 2006, from http://www.wired.com/news/print/0,1294,60588,00.html

Balfanz D., Durfee, G., Grinter, R. E., Smetters, D., & Stewart, P. (2004). Network-in-a-box: How to set up a secure wireless network in under a minute. *Proceedings of the 13th USENIX Security Symposium*, 2004, 207–222.

Balfanz, D., Durfee, G., & Smetters, D. K. (2005). Making the impossible easy: Usable PKI. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 319–334). Sebastopol, CA: O'Reilly.

Balfanz D., Durfee, G., Smetters, D. K., & Grinter, R. E. (2004, September/October). In search of usable security: Five lessons from the field. *IEEE Security and Privacy, 2*(5), 19–24.

Ball, E. (2003). Patient privacy in electronic prescription transfer. *IEEE Security and Privacy, 1*(2), 77–80.

Baumer, D., Earp, J. B. & Payton, F. C. (2000, December). Privacy in medical records: IT implications of HIPAA. *Computers and Society*, 40–47.

Beckles, B., Welch, V., & Basney, J. (2005). Mechanisms for increasing usability of grid security. *International Journal of Human-Computer Studies, 63*(1/2), 74–101.

Begole, J., Tang, J. C., Smith, R. B., & Yankelovich, N. (2002). Work rhythms: Analyzing visualizations of awareness histories of distributed groups, In *Proceedings of CSCW 2002*, 334–343.

Bellotti, V. (1996). What you don't know can hurt you: Privacy in collaborative computing. In *Proceedings of the HCI Conference on People and Computers XI.*

Bellotti, V., & Sellen, A. (1993). Designing for privacy in ubiquitous environments. *Proceedings of the 1993 European Conference on CSCW*, Kluwer, Milan, 77–92.

Bernaschi, M., Gabrielli, E., & Mancini, L. (2000). Operating system enhancements to prevent the misuse of system calls. *Proceedings of the 7th ACM Conference on Computer and Communications Security*, 174–183. Athens, Greece: ACM Pres.

Bhattacherjee, A. (2002). Individual trust in online firms: Scale development and initial trust. *Journal of Management Information Systems, 19(*1), 213–243.

Briggs, P., Buford, B., De Angeli, A., & Lynch, P. (2002). Trust in online advice. *Social Science Computer Review, 20*(3), 321–332.

Briggs, P., Simpson, B., & De Angeli, A. (2004). Personalization and trust: A reciprocal relationship? In C. M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce* (pp. 39–56). Dordrecht, Netherlands: Kluwer Academic Publishers.

Brodie, C., Karat, C-M., & Karat, J. (2004). Personalizing interaction. In C.M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce* (pp. 185–206). Dordrecht, Netherlands: Kluwer Academic Publishers.

Brodie, C., Karat, C., & Karat, J. (2006). An empirical study of natural language parsing accuracy of privacy policy rules using the SPARCLE Policy Workbench. *Proceedings of the Symposium on Usable Privacy and Security,* 8–19. ACM Digital Library.

Brostoff, S., & Sasse, M. A. (2000). Aare passfaces, more usable than passwords? A field trial investigation. In *Proceedings of HCI 2000,* 405–424. Berlin, Germany: Springer.

Buffet, S., Fleming, M. W., Richter, M. M., Scott, N., & Spencer, B. (2004). Determining Internet users' values for privacy information. *Proceed-*

ings of the Second Annual Conference on Privacy, Security, and Trust, *2004*, 79–88.

Caloyannides, M. A. (2004, September/October). Speech privacy technophobes need not apply. *IEEE Security and Privacy, 2*(5), 86–87.

Camenisch, J., & Van Herreweghen, E. (2002). Design and implementation of the Idemix Anonymous Credential System. In *9th ACM Conference on Computer and Communications Security*, 2002, 21–30, New York: ACM Press.

Chaiken, S. (1980). Heuristic versus systematic information processing and the use of source versus message cues in persuasion. *Journal of Personality and Social Psychology, 39*, 752–766.

Conti, G., Ahamad, M., & Stasko, J. (2005). Attacking information visualization system usability: Overloading and deceiving the human. *Proceedings of the Symposium on Usable Privacy and Security*, ACM Digital Library, 89–100.

Corritore, C., Kracher, B., & Wiedenbeck, S. (2003). Online trust: Concepts, evolving themes, a model. *International Journal of Human-Computer Studies, 58*(6), 737–758.

Coventry, L. (2005). Usable biometrics. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 175–198). Sebastopol, CA: O'Reilly.

CRA Conference on Grand Research Challenges in Information Security and Assurance. (2003, November). http://www.cra.org/Activities/grand.challenges/security, 16–19.

Cranor, L. (2002). *Web privacy with P3P*. Sebastopol, CA: O'Reilly Media.

Cranor, L. (2004). 'I didn't buy it for myself': Privacy in ecommerce personalization. In C.M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce* (pp. 57–74). Dordrecht, Netherlands: Kluwer Academic Publishers.

Cranor, L. (2005). Privacy policies and privacy preferences. In L. Cranor & S. Garfinkel (Eds.), *Security and Usability: Designing Secure Systems That People Can Use* pp. 447–472. Sebastopol, CA: O'Reilly.

Culnan, M. (2000). Protecting privacy online: Is self-regulation working? *Journal of Public Policy and Marketing*, 19, 1, 20–26.

Davis, D., Monrose, F., & Reiter, M. (2004). On user choice in graphical password schemes. *Proceedings of the 13th USENIX Security Symposium, 2004,* 151–164.

De Angeli, A., Coventry, L., Johnson, G., & Renaud, K. (2005). Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human Computer Studies, 63*(1/2), 128–152.

de Paula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D., Ren, J., Rode, J., & Filho, R. (2005). In the eye of the beholder: A visualization-based approach to information systems security. *International Journal of Human-Computer Studies, 63*(1/2), 5–24.

Dewan, P., & Shen, H. (1998). Flexible meta access control for collaborative applications. *Proceedings of the 1998 ACM Conference on CSCW*, 247–256, Seattle: ACM Press.

Dey, A. K., & Abowd, G. D. (2000). CybreMinder: A context-aware system for supporting reminders, In *Proceedings of the International Symposium on Handheld and Ubiquitous Computing*, *2000*, 172–186.

Dhamija, R., & Perrif, A. (2000). Deja Vu: A user study using images for authentication. Proceedings of the Ninth USENIX Security Symposium.

Dhamija, R., & Tygar, J. D., (2005). The battle against phishing: Dynamic security skins. In *Proceedings of the Symposium on Usable Privacy and Security,* 77–78. ACM Digital Library.

Dingledine, R., & Matherwson, N. (2005) Anonymity loves company: Usability and the network effect. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use,* 547–560. Sebastopol, CA: O'Reilly.

Dourish, P. (1993). Culture and control in a media space. P*roceedings of the 1993 European Conference on CSCW*, 125–137. Milan,: Kluwer.

Fogg, B. J. (2003). Prominence-interpretation theory: Explaining how people assess credibility online. *Proceedings of the Conference on Human Factors in Computing Systems* (CHI 2003), 722–723. New York: ACM Press.

Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., et al. (2001). What makes websites credible? A report on a large quantitative study. *Proceedings of the Conference on Human Factors in Computing Systems* (CHI 2001) 61–68. New Yor: ACM Press.

Fogg, B. J., Soohoo, C., Danielson, D. R., Marable, L., Stanford, J., & Tauber, E. R. (2003). How do users evaluate the credibility of websites? A study with over 2,500 participants. *Proceedings of the 2003 conference on Designing for User Experiences,* 1–15. New York: ACM Press.

Friedman, B., Kahn, P. H., & Howe, D. C. (2000). Trust online. *Communications of the ACM, 43*(12), 34–40.

Good, N., & Krekelberg, A. (2003). A study of Kazaa P2P file-sharing. *Proceedings of the Conference on Human Factors in Computing Systems—CHI 2003*, 137–144. New York: ACM Press.

Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., et al. (1999). Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the AAAI*, *1999*, 439–446.

Goodall, J., Ozok, A., Lutters, W., Rheingens, P., & Komlodi, A. (2005) A user-centered approach to visualizing network traffic for intrusion detection. *CHI '05 extended abstracts on Human factors in computing systems* (pp. 1403–1406). New York: ACM Press.

Greenberg, S., & Marwood, D. (1994). Real time groupware as a distributed system: concurrency control and its effect on the user interface. P*roceedings of the 1994 ACM Conference on CSCW*, 207–217. Chapel Hill, NC: ACM Press.

Guttmann, P. (2003) Plug-and-Play PKI: A PKI your mother can use. *Proceedings of the 12th USENIX Security Symposium*, 45–58.

Hagan, P. R. (2000). Personalization versus privacy. *Forrester Report*, November, 2000.

Harper, J., & Singleton, S. (2001). *With a grain of salt: What consumer privacy surveys don't tell us*. Retrieved 2006 from, http://ssrn.com/abstract=299930.

Henning, R. (1999). Security service level agreements: quantifiable security of the enterprise? *Proceedings of the 1999 Workshop on New Security Paradigms,* 54–60. Caledon Hills, Canada: ACM Press.

Herbsleb, J. D., Atkins, D. L., Boyer, D. G., Handel, M., & Finholt, T. A. (2002). Introducing instant messaging and chat in the workplace. *Proceeding of the Conference on Human Factors in Computing Systems—CHI 2002*, 171–178. New York: ACM Press.

Hochheiser, H., Bederson, B., Johnson, J., Karat, C., & Lazar, J. (2005). The need for usability of electronic voting systems: Questions for voters and policy makers. *National Academy of Sciences, National Research Council Committee on Electronic Voting.*

Hong, J. I., Ng, J. D., Lederer, S., & Landay, J. (2004). Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *Proceedings of the Conference on Designing Interactive Systems—DIS'04*, 91–100. New York: ACM Press.

Hong, J. I., & Landay, J. (2004). An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of MobiSys'04*, 177–189. New York: ACM Press.

Horstmann, T., & Bentley, R. (1997). Distributed authoring on the web. with the BSCW shared workspace system.*ACM Standards View*, *5,* 9–16.

Irvine, C., & Levine, T. (1999). Toward a taxonomy and costing method for security services. *Proceedings of the 15th Computer Security Applications Conference,* 183–188. Phoenix, AZ: IEEE.

Jensen, C., & Potts, C. (2004). Privacy policies as decision-making tools: A usability evaluation of online privacy notices. *Proceedings of the Conference on Human Factors in Computing Systems—CHI 2004* (pp. 471–478). New York: ACM Press.

Jensen, C., Potts, C., & Jensen, C. (2005). Privacy practices of Internet users: Self-reports versus observed behavior. *International Journal of Human-Computer Studies, 63*, 203–227.

Jøsang, A., & Presti, S. L. (2004, April). Analysing the relationship between risk and trust. In T. Dimitrakos (Ed.), *Proceedings of the Second International Conference on Trust Management*, Oxford, 135–145. Berlin: Springer-Verlag.

Just, M. (2004, September/October). Designing and evaluating challenge-question systems. *IEEE Security and Privacy, 2*(5), 32–39.

Kahn, D. (1967). *The codebreakers: The story of secret writing*. MacMillan, New York, NY.

Kandogan, E., & Haber, E. (2005). Security administration tools and practices. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use,* 357–378. Sebastopol, CA: O'Reilly.

Karat, C., Brodie, C., & Karat, J. (2005) Usability design and evaluation for privacy and security solutions. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use,* 47–74. Sebastopol, CA: O'Reilly.

Karat, C., Karat, J., Brodie, C., & Feng, J. (2006). Evaluating interfaces for privacy policy rule authoring. *Proceedings of the Conference on Human Factors in Computing Systems—CHI 2006.* ACM Press.

Karat, J., Karat, C., & Brodie, C. (2004). Personalizing interaction. In C.-M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce* 7–18. Dordrecht, Netherlands: Kluwer Academic Publishers.

Karat, J., Karat, C., Brodie, C., & Feng, J. (2005). Privacy in information technology: Designing to enable privacy policy management in organizations. *International Journal of Human Computer Studies, 63*(1/2), 153–174.

Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (1998). Cryptanalytic attacks on psedudorandom number generators. *Proceedings of the Fifth International Workshop on Fast Software Encryption*, 168–188 Berlin: Springer.

Kemmerer, R., Meadows, C., & Millen, J. (1994). Three systems of cryptographic protocol analysis. *Journal of Cryptography, 7*(2), 79–130.

Kobsa, A. (2002). Personalized hypermedia and international privacy. *Communications of the ACM, 45*(5), 64–67.

Kuhlen, R. (1998, October 1-3). *Trust: A principle for ethics and economics in the global information society*. Paper presented at *The Second UNESCO Congress for Informational Ethics*, Monte Carlo, Monaco.

Kumaraguru, P., & Cranor, L. (2005). Privacy in India: Attitudes and awareness. In *Proceedings of the 2005 Workshop on Privacy Enhancing Technologies* (*PET2005*), 1–17 30 May.

Lederer, S., Mankoff, J., & Dey, A. K. (2003). Who wants to know what when? Privacy preference determinants in ubiquitous computing. In *Proceedings of the Conference on Human Factors in Computing Systems—CHI—2003,* 724–725. ACM Press.

Malin, B., & Sweeney, L. (2004). How (not) to protect genomic data privacy in a distributed network: Using trail re-identification to evaluation and design anonymity protection systems. *Journal of Biomedical Informatics, 37*(3), 179–192.

Mandler, G. (1991). Your face is familiar but I can't remember your name: A review of dual process theory, relating theory and data. *Journal of Experimental Psychology, Learning, Memory, and Cognition, 8*, 207–225.

Manny, C. H. (2003). European and American privacy: Commerce, rights, and justice. *Computer Law and Security Report*, *19*(1), 4–10.

Mantei, M. M., Baecker, R. M., Sellen, A. J., Buxton, W. A. S., Milligan, T., & Wellman, B. (1991). Experiences in the use of a media space. In *Proceedings of the Conference on Human Factors in Computing Systems—CHI'91,* 203–208. New York: ACM Press.

Marsh S., & Dibben M. (2003). The role of trust in information science and technology. In B. Cronin (Ed.). *Annual Review of Information Science and Technology,* 37, 465–498.

Maxion, R., & Reeder, R. W. (2005, July). Improving user interface dependabiity through mitigation of human error. *International Journal of Human-Computer Studies. 63*(1–2), 25–50.

Meyer, S., & Rakotonirainy, A. (2003). A survey of research on context-aware homes. *Proceedings of the Australasian Information Security Workshop Conference on ACSW frontiers 2003, 21*, 159–168.

Milberg, S. J., Burke, S. J., Smith, H. J., & Kallman, E. A. (1995). Values, personal information privacy, and regulatory approaches. *Communications of the ACM, 38*(12), 65–74.

Miller, R., & Wu, M. (2005). Fighting phishing at the user interface. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 275–292). Sebastopol, CA: O'Reilly.

Monrose, F., & Reiter, M. (2005). Graphical password systems. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* 157–174. Sebastopol, CA: O'Reilly.

National Research Council. (2003). *Who goes there? Authentication through the lends of privacy*. Washington, DC: National Academies Press.

OECD (1980). *OECD guidelines on the protection of privacy and transborder flows of personal data.* http://www.oecd.org/home/

Office of the Federal Privacy Commissioner of Australia. (2000). *Privacy and Business*. http://www.privacy.gov.au

Office of the Federal Privacy Commissioner (2001, July). *Privacy and the community*. Retrieved 2006, published survey research available at www.privacy.gov.au.

Palen, L., & Dourish, P. (2002). Unpacking 'privacy' for a networked world. *Proceedings of the Human Factors in Computing Systems Conference (CHI 2002)*, 129–136. ACM Press.

Palmer, J. W., Bailey, J. P., & Faraj, S., (2000). The role of intermediaries in the development of trust on the WWW: The use and prominence of trusted third parties and privacy statements. *Journal of Computer-Mediated Communication, 5*(3). Retrieved 2006, from http://jcmc.indiana.edu/vol5/issue3/palmer.html.

Patrick, A., Briggs, P., & Marsh, S. (2005). Designing systems that people will trust. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 75–100). Sebastopol, CA: O'Reilly.

Peacock, A., Ke, X., & Wilkerson, M. (2005). Identifying users from their typing patterns. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 199–220). Sebastopol, CA: O'Reilly.

PET: The workshop on Privacy Enhancing Technologies. (2005). Retrieved in 2006 at http://petworkshop.org/2005/).

Pettersson, J. H., Fischer-Huebner, S., Danielsson, N., Nilsson, J., Bergmann, M., Clauss, S., et al. (2005). Making PRIME usable. *Proceedings of the Symposium on Usable Privacy and Security*, 53–64. ACM Digital Library.

Piazzalunga, U., Salvaneschi, P., & Coffetti, P. (2005). The usability of security devices. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 221–244) Sebastopol, CA: O'Reilly.

Real User Corporation. (2002). *The science behind passfaces*. Retrieved 2006, from http://www.realuser.com/published/ScienceBehindPassfaces.pdf

Reiter, M., & Rubin, A. (1998). Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 66–92.

Renaud, K. (2005) Evaluating authentication mechanisms. In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 103–128). Sebastopol, CA: O'Reilly.

Salzter, J., & Schroeder, M. (1975). The protection of information in computer systems. In *Proceedings of IEEE, 63*(9), 1278–1308.

Sasse, A., & Flechais, I. (2005). Usable security: Why do we need it? How do we get it? In L. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use,* 13–30. Sebastopol, CA: O'Reilly.

Schneier, B. (2000). *Secrets and lies: Digital security in a networked world*. New York, NY: Willey Computer Publishing.

Schneier, B., & Mudge, D. (1998). Cryptanalysis of Microsoft's point-to-point tunneling protocol (PPTP). *Proceedings of the 5th ACM Conference on Computer and Communications Security*, 132–141. San Francisco, CA: ACM Press.

Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, L., Ekin, A., Connell, J., Shu, C., & Lu, M. (2005). Enabling video privacy through computer vision. I*EEE Security and Privacy, 3*(3), 50–57.

Shankar, U., Talwar, K., Foster, J., & Wagner, D. (2002). Detecting format string vulnerabilities with type qualifiers. In *Proceedings of the 10th USENIX Security Symposium*, Washington, DC, 201–220.

Shen, H., & Dewan, P. (1992). Access control for collaborative environments. *Proceedings of the 1992 ACM Conference on CSCW*, 51–58. Toronto, Canada: ACM Press.

Shubina, A. M., & Smith. S. W. (2004). Design and prototype of a coercion-resistant, voter verifiable electronic voting system. In *Proceedings of the Second Annual Conference on Privacy Security, and Trust*, 2004, 29–39.

Sillence, E., Briggs, P., Fishwick, L., & Harris, P. (2004). Trust and mistrust of online health sites. *Proceedings of the Conference on Human Factors in Computing Systems—CHI 2004*, 663–670. New York: ACM Press.

Smith, J. (1993). Privacy policies and practices: Inside the organizational maze. *Communications of the ACM, 36*(12), 105–122.

Spiekermann, S., Grossklags, J., & Berendt, B. (2001). E-privacy in 2nd generation e-commerce: Privacy preferences versus actual behavior. In *Proceedings of the ACM conference on Electronic Commerce*, 38–47.

Spyropoulou, E., Levin, T., & Irvine, C. (2000). Calculating costs for quality of security service. *Proceedings of the 16th Computer Security Applications Conference*, 334–343. New Orleans, LA: IEEE.

Sweeney, L. (2001). k-Anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10*(5), 557–570.

Teltrow, M., & Kobsa, A. (2004). Impacts of user privacy preferences on personalized systems. In C. M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce*, 315–332. Dordrecht, Netherlands: Kluwer Academic Publishers.

Thomsen, D., & Denz, M. (1997) Incremental assurance for multilevel application. *Proceedings of the 13th Computer Security Applications Conference*, 81–89. San Diego, CA: IEEE.

Thuraisingham, B. (2002). Data mining, national security, privacy and civil liberties. *ACM SIGKDD Explorations Newsletter, 4*(2), 1–5.

Turow, J. (2003). *Americans and Online Privacy: The System is broken*. Philadelphia: Annenberg Public Policy Center.

U.S. Fair Credit Reporting Act of 1978. U.S. Code 1681. www.ftc.gov/os/statutes/031224fcra.pdf.

U.S. Fair and Accurate Credit Transaction Act. H.R. 2622, 108th Congress, July 24, 2003.

Wagner, D., Foster, J., Brewer, E., & Aiken, A. (2000). A first step toward automated detection of buffer overrun vulnerabilities. In *Network Distributed Systems Security Symposium*, pp. 1–15, Feb. 2000.

Want, R., Hopper, A., Falcão V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems, 10*(1), 91–102.

Warren, S. A., & Brandeis, L. D. (1890, December). The right to privacy. *Harvard Business Review, 4*, 195.

Welty, B., & Becerra-Fernandez, I. (2001). Managing trust and commitment in collaborative supply chain relationships. *Communications of the ACM, 44*(6), 67–73.

Whitten, A., & Tygar, D. (1999). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, 1999, 169–184.

Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., & Memon, N. (2005). PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, *63*, 102–127.

Yan, J., Blackwell, A., Anderson, R., & Grant, A. (2004). Password memorability and security: Empirical results. *IEEE Security and Privacy, 2*(5), 25–31.

Yee, K. (2005). Guidelines and strategies for secure interaction design. In L. Cranor & S. Garfinkel, (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 247–273). Sebastopol, CA: O'Reilly.

Yenisey, M., Ozok, A., & Salvendy, G. (2005) Perceived security determinants in e-commerce among Turkish university students. *Behaviour and Information Technology, 24*(4), 259–274.

Zhang, J., & Ghorbani, A. A. (2004). Familiarity and trust: Measuring familiarity with a website. *Proceedings of the Second Annual Conference on Privacy Security, and Trust*, 2004 23–28.

Zurko, M. E., & Simon, R. T. (1996). *User-centered security*. Paper presented at the New Security Paradigms Workshop.

Part

IV

APPLICATION/DOMAIN SPECIFIC DESIGN

·33·

HUMAN-COMPUTER INTERACTION IN HEALTH CARE

François Sainfort and Julie A. Jacko
Georgia Institute of Technology

Paula J. Edwards
Georgia Institute of Technology & Children's Health Care of Atlanta

Bridget C. Booske
University of Wisconsin-Madison

Introduction **662**
Characteristics of the Healthcare Industry **662**
The Healthcare Industry 662
Complexity of Science and Technology 662
Chronic Conditions 663
Organization of the Delivery System 663
Information Technology 663
Consumerism 663
The Healthcare Regulatory Environment 664
Transaction Standards and Coding Sets 664
Privacy and Security 664
Healthcare Informatics **665**
Consumer Health Information 665
Interactive Health Communication 665
Consumer/Patient Web-Based Applications and e-Health 666
Provider Healthcare Informatics 666
Electronic Medical Records EMRs 666
Computerized Physician Order Entry Systems 667
Patient Monitoring Systems 667
Imaging Systems 667
Information-Retrieval Systems 668
Decision-Support Systems 668
Clinical- Decision-Support Systems 668
Patient Decision-Support Systems 668
Patient Preferences 669
HCI Challenges in Patient DSS 669
Evaluating Computerized Healthcare Applications **669**
Future Opportunities and Challenges **674**
Supporting/Enhancing the Patient-Provider Interaction 674
Supporting/Enhancing Collaborative Work Among Providers 675
Developing and Utilizing New Information and Communication Technologies 675
Designing and Utilizing Adaptive HCIs 675
Moving to e-Health 675
References **676**

INTRODUCTION

U.S. healthcare expenditures were nearly $1.7 trillion in 2003 and were expected to grow to $2 trillion by 2005 (BlueCross BlueShield Association). Despite such large spending, many Americans remain uninsured and do not have access to healthcare services. Furthermore, while our country has the most formidable medical workforce in the world and develops and uses the most modern medical technologies, the World Health Organization (2000) recently rated the quality and performance of the U.S. healthcare systems as being worse than most of its counterparts in the western world. Chassin, Galvin, and the National Roundtable on Health Care Quality (1998) documented three types of quality problems: (a) overuse, (b) underuse, and (c) misuse. The results of an extensive review of over 70 publications covering years 1993 through 2000 provide "abundant evidence that serious and extensive quality problems exist throughout American medicine resulting in harm to many Americans" (Institute of Medicine, 2001, p. 24). In its first report, *To Err is Human*, the Institute of Medicine (2000) reported serious and widespread errors in healthcare delivery that resulted in frequent avoidable injuries to patients. The Institute of Medicine (2001) suggested four key underlying reasons for inadequate quality of care in the U.S. healthcare system: (a) the growing complexity of science and technology, (b) the increase in chronic conditions, (c) a poorly organized delivery system, and (d) constraints on exploiting the revolution in information technology.

Reengineering the delivery of healthcare services through innovative development, application, and use of information and medical technologies can result in tremendous cost savings, improved access to healthcare services, as well as improved quality of life for all citizens. Healthcare professionals in the United States have recognized that both the information revolution and the biological revolution will offer tremendous opportunities—and challenges—for (re)designing the healthcare system of the future. They are aware of the need to better utilize new information and communication technologies and incorporate computing power into care delivery and clinical practice. They are also aware that the widely publicized biological revolution (which includes both advances in genetics and advances in biomedical engineering) will soon bring a large number of screening and diagnostic tests as well as new treatment strategies and disease-management tools. It is clear that the combination of biotechnology, computing power, information and communication technologies, distance technology, and sensor technology will make future delivery of healthcare in the United States unrecognizable from the care we deliver today.

However, unlike other parts of the American economy, the healthcare system has been slow to embrace modern information and communication technology, which is often viewed as too expensive, very unusable, and quite divergent from current practice (Schoen, Davis, Osborn, & Blendon, 2000). In addition, technologies do not typically come "ready-to-use" off the shelf and it is extremely difficult, if not impossible, for healthcare provider organizations to have the range of expertise, in house, that is needed to design, adapt, and implement technologies to meet an organization's needs. In fact, many healthcare organizations are often not even fully aware of their own needs, do not know which technologies are available for what, and do not know how modern information and communication technologies can be effectively used to improve and simplify care delivery. In many cases, recent information technology adoptions among healthcare providers have been driven by Federal and state regulations and requirements rather than well-recognized internal needs and growth opportunities. For example, public and private groups (i.e., the federal government and the Leap Frog group, a collaboration of businesses) are strongly encouraging implementation of Computerized Physician Order Entry (CPOE) and other information technologies in order to achieved improvements in the quality and efficiency of providing care (Kaushal, Shojania, et al. 2003).

Healthcare informatics is a field that can be widely defined as the generation, development, application, and testing of information and communication principles, techniques, theories, and technologies to improve healthcare delivery. It includes the understanding of data, information, and knowledge used in the delivery of healthcare and an understanding of how these data are captured, stored, accessed, retrieved, displayed, interpreted, used and made more efficient. While healthcare informatics intersects with the field of medical or bioinformatics, it is different in the sense that it focuses on healthcare delivery, and hence is centered on the patient (and/or consumer), the clinician (or healthcare professional, or "provider"), and, more importantly, the patient-provider interaction. Human-computer interaction (HCI), from the perspective of both the patient/consumer as well as the provider, is essential to the success of "healthcare informatics." In this chapter, we first review the characteristics of the healthcare industry in the United States. We then review information systems used by consumers, patients, and providers and raise HCI issues and challenges associated with both perspectives. Then we propose a framework for evaluating healthcare applications and conclude with a discussion of future opportunities and challenges for HCI in healthcare.

CHARACTERISTICS OF THE HEALTHCARE INDUSTRY

The Healthcare Industry

As previously discussed, the Institute of Medicine (2001) puts forth four key underlying reasons for inadequate quality of care in the U.S. healthcare system today: (a) the growing complexity of science and technology, (b) the increase in chronic conditions, (c) a poorly organized delivery system, and (d) constraints on exploiting the revolution in information technology. In addition, a growing trend toward consumerism is becoming a major force in shaping the future organization of the healthcare industry. These five trends, detailed further in the following discussion, are shaping the future of healthcare in the United States.

Complexity of Science and Technology

The sheer volume of new healthcare science and technologies—the knowledge, skills, interventions, treatments, drugs,

and devices—is very large today and has advanced much more rapidly that our ability to use and deliver them in a safe, effective, and efficient way. Government as well as private investment in pharmaceutical, medical, and biomedical research and development has increased steadily. The healthcare delivery system has not kept up with phenomenal advancement in science and technology and proliferation of knowledge, treatments, drugs, and devices. With current advances in genomics (offering promise in diagnosis as well as, possibly, treatment), sensor technologies (offering promise in automated detection, measurement, and monitoring), nanotechnologies (offering promise in diagnosis, treatment, and control), and information and communication technologies (enabling remote delivery, telemedicine, e-health, and patient empowerment), the complexity of science and technology in healthcare is only going to increase.

Chronic Conditions

As noted by the Institute of Medicine (2001), "because of changing mortality patterns, those age 65 and over constitute an increasingly large number and proportion of the U.S. population" (p. 26). Therefore, there is an increase in both the incidence and prevalence of chronic conditions (defined as conditions lasting more than three months and not self-limiting). Hoffman, Rice, and Sung (1996) estimated that patients with chronic conditions make up 80% of all hospital bed days, 83% of prescription drug use, and 55% of emergency-room visits. Compared to acute illnesses, effectively treating chronic conditions requires disease management and control over long periods of time, collaborative processes between providers and patient, as well as patient involvement, self-management, and empowerment.

Organization of the Delivery System

The healthcare delivery system in the United States is a highly complex system that is nonlinear, dynamic, and uncertain. The system is further complicated by a large number of agents who are multiple stakeholders, each with multiple, sometimes conflicting, goals, aspirations, and objectives. As a result, the entire system leads to a lack of accountability; it is has frequently misaligned reward as well as incentive structures, and it suffers from inefficiencies embedded in multiple layers of processes. The healthcare "product" or "service" is often ill defined or difficult to define and evaluate. The processes involved in delivering healthcare services are complex, ill specified, and difficult to measure, monitor, and control. Health outcomes are difficult to measure, manage, and analyze. The system experiences growing cost pressures, faces potential insurance premium increases, and is extremely fragmented. Wagner, Austin, and Von Korff (1996) identified five elements needed to improve patients' outcomes in a population increasing afflicted by chronic conditions:

- evidence-based, planned care
- reorganization of practices to meet the needs of patients who require more time and/or resources, and closer follow-up
- systematic attention to patients' need for information and behavioral change
- ready access to necessary clinical knowledge and expertise
- supportive information systems

Regarding this last point, the Institute of Medicine (2001) pointed to the fact that:

> healthcare organizations are only beginning to apply information technology to manage and improve patient care. A great deal of medical information is stored on paper. Communication among clinicians and with patients does not generally make use of the Internet or other contemporary information technology. Hospitals and physician groups operate independently of one another, often providing care without the benefit of complete information on the patient's condition or medical history, services provided in other settings, or medications prescribed by other providers. (p. 30)

Information Technology

The revolution in information technology holds great promise in a number of areas for consumers, patients, clinicians, and all organizations involved in the delivery of healthcare services. A recent report by the National Research Council of The National Academies (2000) identified six major information technology applications domains in healthcare: (a) consumer health, (b) clinical care, (c) administrative and financial transactions, (d) public health, (e) professional education, and (g) research. While many applications are currently in use (such as online search for medical information by patients), others, such as remote and virtual surgery and simulation of surgical procedures, are in early stages of development (Institute of Medicine, 2001). While the Internet (and intranets) has been a driving force for changes in the information technology landscape in the past decade, many healthcare applications are not web based. Applications such as administrative billing systems, electronic medical records EMR), and computerized physician order entry systems frequently remain on legacy systems, which are often built around older mainframe systems and lacking integration with other applications.

With respect to consumers/patients and providers, the Committee on the Quality of Health Care in America identified five key areas in which information technology could contribute to an improved delivery system (Institute of Medicine, 2001):

- access to medical knowledge base
- computer-aided decision support systems
- collection and sharing of clinical information
- reduction in medical errors
- enhanced patient and clinician communication

Consumerism

The Internet and other developments in information and communication technologies are contributing to greater consumerism with stronger demands from individuals for information and convenience. People are more demanding; they want

timely and easy access to medical information, the latest in technology, and the latest in customer service. Patients are starting to have access to tools that can lead to empowerment and shared decision making regarding their own healthcare. In addition, because of increased fragmentation and specialization in medical care, patients need to take a more active role in managing their health and healthcare to ensure that the various providers involved in providing their care (i.e., primary care physician, specialist, pharmacist, etc.) have the information they need to provide appropriate, quality care.

There are, however, many technical, organizational, behavioral, and social challenges and barriers to a increased use of information technology by individuals in managing their health/healthcare. These technological challenges not only include the design of optimal, effective, flexible human-computer interfaces (HCIs), but also issues of privacy and security of information. Those are discussed in the following section.

The Healthcare Regulatory Environment

As Kumar and Chandra (2001) mentioned, the healthcare industry has unique legislative challenges. Among them, two in particular have implications on the field of healthcare informatics: (a) the Health Insurance Portability and Accountability Act (HIPAA); and (b) health information and other business data security. The Health Insurance Portability and Accountability Act was passed and signed into law in 1996 and is designed to improve the portability of health-insurance coverage in the group and individual markets, limit healthcare fraud and abuse, and simplify the administration of health insurance. The act has serious, impending implications for healthcare providers and information managers. Of all its mandates, administrative simplification is perhaps the most critical for healthcare information managers, who are faced with everything from establishing standardized financial and clinical electronic data interchange (EDI) code sets to adopting, assigning, and using unique numerical identifiers for each healthcare provider, payer, patient, and employer. Both HIPAA and the growing role of the Internet-based technologies in delivering healthcare create an even greater and critical concern for data security and privacy.

HIPAA contains five key Titles that apply to every provider, payer (including self-insured employers), and healthcare clearinghouses. Title I pertains to healthcare access, portability, and renewability. Title II relates to healthcare fraud and abuse. Title III addresses tax-related provisions and medical savings accounts. Title IV discusses the application and enforcement of group health-plan requirements. Lastly, Title V relates to revenue offsets.

Currently, the healthcare industry is focusing its efforts on Title II of the Act, also known as Administrative Simplification. Unlike past legislation that has imposed regulations on only Medicare and Medicaid, all healthcare providers, payers, and clearinghouses must meet HIPAA Administrative Simplification regulations within 26 months of each subsection's finalization. There are four subsections to Title II. They are:

- transaction standards
- coding sets
- patient privacy
- security

Transaction Standards and Coding Sets

The Department of Health and Human Services has adopted national standards for electronic administrative and financial healthcare transactions. This is one of the most positive attributes of Title II as it eliminates the conflicting transaction standards, coding sets, and identifiers used by the various players in the industry. Initial transaction and code sets were implemented in October 2002, with a revised code set standards defined and implemented by October 2003. The standards relate to enrollment, referrals, claims, payments, eligibility for a health plan, payment and remittance advice, premium payments, health-claim status, and referral certifications and authorizations. By developing national standards, it is anticipated that Electronic Data Interchange (EDI) of healthcare data will significantly reduce administrative costs. The Workgroup on Electronic Data Interchange (2000) estimated that EDI can reduce administrative costs by $26 billion per year by streamlining precertification, enrollment status, and reimbursement processes. Code Sets have also been defined in which variations are not permitted. Code Sets are based on the following standards:

ICD-9 Volumes 1 & 2—Diagnosis Coding
ICD-9 Volume 3—Inpatient Hospital Service Coding
CPT 4—Physician Service Coding
CDT-2—Dental Service Coding
HCPCS—Other Health Related Coding
DRG—Diagnosis Related Groups
NDC—National Drug Coding

HIPAA also establishes unique identifiers for healthcare providers, health plans and payers, employers, and eventually patients/individuals. Currently the patient/individual identifier is pending as no consensus could be reached.

Privacy and Security

Administrative simplification also addresses patient privacy and security. Privacy standards exist for disclosure of patient identifiable information (including demographic data), training of healthcare workforce, individual's rights to see records, procedures for amending inaccuracies in medical records, maintenance of privacy when patient information is exchanged between business associates, designation of a privacy officer, procedures for complaints, sanctions for infractions, duty to mitigate, and document compliance. As a component of disclosure, covered entities (providers, payers, and clearinghouses) must make reasonable efforts not to use or disclose more than the minimum amount of protected patient information necessary to accomplish the intended purpose of the use of disclosure. Wide adoption and use of electronic medical record (EMR) systems will present technical challenges to preventing disclosure of this information. In the past, accessibility of paper-based patient records to clinicians was restricted in part by the chart's physical location (i.e., in medical records storage, on the nursing unit in a hospital). However, EMR makes these charts available to a much large number of clinicians in a wider range of physical

locations, making it more difficult to prevent inappropriate uses of this protected information by authorized users of the EMR.

Security standards require establishment of administrative procedures (policies and procedures), physical safeguards (physical access to computers), technical security (individual and network computer access), and electronic signature (optional, but if used, it must be digital). Implementation requirements of the privacy and security standards have received a tremendous amount of debate. While most feel the intent of the standards is good, the cost implications have left many advocating for a longer implementation period. The Department of Health and Human Services estimated the total cost to implement Subtitle II to be approximately $17 billion to $18 billion over the next 10 years. Some industry experts (Kennedy and Blum, 2001) purported implementation and maintenance costs ranging from $50 to $200 million for a large healthcare provider and/or payer. However, these numbers may increase as more healthcare providers implement wireless networks to support the use of EMR and other clinical systems. Use of wireless networks and centralized databases that support EMR and other systems creates a new set of technical security requirements and issues.

HEALTHCARE INFORMATICS

As defined earlier, healthcare informatics comprises the generation, development, application, and testing of information and communication principles, techniques, theories, and technologies to improve the delivery of healthcare with a focus on the patient/consumer, the provider, and, more importantly, the patient-provider interaction. While a number of systems using a number of platforms and technologies have already been developed and are currently being developed, the field itself is still in its infancy. In addition, current systems have been designed to fit within the existing healthcare delivery system and thus are only marginally or superficially impacting the way healthcare is being delivered. The true potential of healthcare informatics is yet to be experienced and will radically transform the way healthcare is being delivered and managed in the future. The following sections provide background information on current healthcare applications with an emphasis on two types of users: (a) the consumer/patient and (b) the clinician/provider.

Consumer Health Information

Consumer health information has been defined as "any information that enables individuals to understand their health and make health-related decisions for themselves or their families" (Harris, 1995). Patrick and Koss (1995) listed a variety of organizations and entities that produce and/or disseminate consumer health information including health-related organizations (involved in provision of, or payment for, healthcare services and supporting services), libraries, health voluntary organizations (i.e., American Heart Association, American Cancer, American Lung, etc., and 6000+ other health-interest societies), broadcast and print media, employers, government agencies, community-based organizations (i.e., churches, YMCA, agencies for the elderly), networked computer health information providers—"virtual" communities. Methods of dissemination are diverse and include informal channels, printed text, broadcast electronic media, dial-up services (telephone), nonnetworked computer-based information (i.e., CD-ROM, Kiosk technology), and networked interactive computer-based information. Harris (1995) discussed a number of problems with health information including how to interpret conflicting or differing information; how to judge reliability; how to choose among many alternatives; and how to deal with the vast quantities of information, much of which is superficial or even inaccurate.

Interactive Health Communication

Print and broadcast dissemination of health information leads to a number of problems such as the timing relative to need, single directionality (difficulty of following up, clarifying, and understanding), timeliness and relevance vis-à-vis updates, and not being unique to individuals (Patrick & Koss, 1995). As Harris (1995) pointed out, electronic sources of health information have the potential to be more timely and complete than other media and can become more accessible to all citizens. Consequently, the use of interactive health communication or consumer health informatics has become increasingly popular. Unfortunately, the advantages have not been fully realized. The science panel on interactive communication and health defined interactive health communication as "the interaction of an individual—consumer, patient, caregiver, or professional—with or through an electronic device or communication technology to access or transmit health information or to receive guidance and support on health-related issues" (Robinson, Patrick, Eng, & Gustafson, 1998, p. 1284). The panel identified six potential functions of interactive health communication applications: (a) relay information, (b) enable informed decision making, (c) promote healthful behaviors, (d) promote peer information exchange and emotional support, (e) promote self-care, and (d) manage demand for health services.

Ferguson (1997) referred to two types of consumer health informatics: community and clinical. Community consumer health informatics includes resources such as online networks, forums, databases, and websites that anyone with a home computer can access. Clinical consumer health informatics includes resources such as programs or systems developed by clinicians, system developers, or HMOs, and is provided only to selected groups of members or patients. With respect to community consumer health informatics resources, there is a growing concern that the barriers to use of online health resources are becoming similar to those for actual health services. These barriers include cost, geography, literacy, culture, disability, and other factors related to the capacity of people to use services appropriately and effectively. Eng et al. (1998) urged public and private sector organizations to collaborate in reducing the gap between the "haves" and "have nots" for health information. They suggested the need for supporting health information technology access in homes and public places, developing applications for the growing diversity of users, funding research on access-related issues, enhancing literacy in health and technology,

training health information intermediaries, and ensuring the quality of health information and support.

In a report on consumer health informatics, the U.S. General Accounting Office (1996) described three general categories of consumer health informatics. They include systems that (a) provide health information to user (one-way communication); (b) tailor specific information to the user's unique situation (customized communication); or (c) allow the user to *communicate and interact* with healthcare providers or other users (two-way communication). The report also cited a number of issues of concern: access, cost, information quality, security and privacy, computer literacy, copyright, systems development (compatibility, infrastructure, and standardization), and the potential for information overload.

Consumer/Patient Web-Based Applications and e-Health

An estimated 95 million Americans seek health information online (Fox 2005). In 1999, the Benton Foundation estimated there were 10,000 or more health-related websites. However, this number has grown tremendously to meet increasing consumer demand for healthcare information. In early 2006, the Yahoo directory of health-related websites (dir.yahoo.com/health) listed over 10,000 websites related to diseases and conditions alone and thousands of other sites addressing topics ranging from pharmacy to alternative medicine to public health. This variety of information available demonstrates that consumers and patients desire a range of information and services including getting disease-treatment information, obtaining report cards on physicians or hospitals, exchanging information with other patients, interacting online with their physicians, and managing their own health benefits. The sheer volume of information on the Internet exceeds most expectations yet raises problems: efficient search for information, information retrieval, information visualization, human-information processing, understanding, and assimilation. Authors of Internet information need to determine how to best structure the information for use by others. E-Health can be described as the transition of healthcare processes and transactions into the Internet-delivered electronic superhighway. Potential problems exist that are specific to patients seeking electronic health-related information (Sonnenberg, 1997), including the lack of editorial control of information, conflict of interest for website sponsors, and unfiltered information presenting an unbalanced view of medical issues. However, referring primarily to self-help groups, Ferguson (1997) believed that the problem of obtaining bad medical information online is not very different from obtaining bad medical information at cocktail parties, in the tabloids, in magazine ads, etc. In addition to the content of information, there are technological problems that may affect both access and use of health information on the Internet: slow modems, poor institutional Internet connections, firewalls that interfere with Internet traffic, malfunctioning message routers, and heavy Internet usage in the immediate geographic area (Lindberg & Humphreys, 1998). Most current concerns about interactive health-communication center on the fact that these applications have the potential to both improve health and to cause harm; thus highlighting the need to ensure their accuracy, quality, safety, and effectiveness (Robinson et al., 1998).

Provider Healthcare Informatics

Providers and healthcare provider organizations have long used health-information systems to support both administrative and clinical functions of healthcare delivery and management. However, despite the fact that healthcare is one of the most information-intensive industries; it has very few state-of-the-art information-management systems. Healthcare is fragmented, with hundreds of thousands of payers, hospitals, physicians, laboratories, medical centers, pharmacies, and clinics, each with its own legacy of systems, hardware, software, and platforms. Electronic data interchange and connectivity issues have become critical. Numerous information systems have been developed and implemented, the most noteworthy being integrated EMRs and computerized physician order entry systems. The following is a discussion of some important clinical applications.

Electronic Medical Records EMRs

EMRs are increasingly being adopted in both primary-care and inpatient-care environments. EMRs provide functions to document all clinical processes and patient-related information relevant to the delivery of patient care. EMRs are advocated and used to improve the quality, accessibility, and timeliness of patient medical information. However, because EMRs replace traditional patient charts, they affect a wide range of clinical users and clinical processes. As such, they face a number of barriers to adoption, including cost, a lack of tested systems, problems with data entry, inexperienced vendors, confidentiality concerns, and security concerns (Wager, Lee, White, Ward, & Ornstein, 2000). In addition, EMR may have barriers that are specifically related to the practice of medicine. For example, physician use of EMR while with patients could affect patient perceptions of quality of care or quality of physician-patient interactions. While some past studies examining this issue have not shown this to be the case (i.e., Legler & Oates, 1993), a recent survey of pediatric-care primary physicians using EMR indicated that they felt it reduced eye contact with patients and increased the duration of patient visits (Adams, Mann, & Baushner, 2003). Thus, the impact of these technologies on physician-patient interactions is not clear at this point.

A key factor that will contribute to whether EMR systems become more rapidly adopted in primary-care practices is whether or not physicians themselves perceive the systems as improving quality (of the medical records, patient care, and overall performance). A recent study examined this issue using qualitative methods in five community-based practices that had used EMR for at least two years and did not use a duplicate record system (Wager et al., 2000). Results indicated that many physicians and staff members perceived benefits, such as increased access, an ability to search the system, and improved overall quality of medical records. There were, however, several disadvantages mentioned, including the frequency of downtime and the time necessary to develop customized templates.

In a recent study examining the quality of worklife of family physicians, Karsh, Beasley, Hagenauer, and Sainfort (2001) collected quantitative data about EMR in order to compare perceptions of medical records between physicians using EMR and those not using EMR. Specifically, they assessed whether or not the use of EMR was related to perceptions of improved quality of medical records. The results showed that physicians who used EMR perceived their medical records to be more up to date and accessible. Physicians who used EMR were also more satisfied with the overall quality of their medical-records systems. On the other hand, there were no differences in perceptions of whether medical records could be modified to meet individual needs. This suggests that EMR can have positive impacts on medical records (it is possible, though, that the positive responses were caused not by positive traits of the EMR, but rather because of cognitive dissonance).

The results of Karsh et al.'s (2001) study supported those of qualitative studies of physicians who use EMR. Wager et al. (2000) found that physicians in primary practice who had used EMR for at least two years believed EMR to have many benefits over paper-based systems, including increased access and availability of patient information to multiple users, the ability to search the system, improved overall quality of patient records, improved quality of documentation, increased efficiency, facilitated cross-training, and improved communication within the practice. Thus, it is clear that EMR has the potential to improve the quality of patient records and therefore, possibly, the quality of care. However, to fully capitalize on such systems, they need to be designed to maximize usability, connectivity, and portability while guaranteeing privacy and security.

Computerized Physician Order Entry Systems

Computerized physician order entry (CPOE) systems have substantial potential for improving the medication ordering process because they enable physicians to write orders directly online. They ensure complete, unambiguous, and legible orders; they assist physicians at the time of ordering by suggesting appropriate doses and frequencies, by displaying relevant data to assist in prescription decisions, by checking drugs prescribed for allergies and drug-drug interactions. As such, those systems are believed to potentially reduce the incidence of medical errors in general, and medication errors in particular. In a study of the impact of CPOE, Bates, Teich, Lee, Seger, Kuperman, Ma' Luf, Boyle, and Leape (1999) found that such systems substantially decreased the rate of nonmissed-dose medication errors. In addition, other studies have demonstrated that use of CPOE decreases turn-around time for medication orders as well as radiation and laboratory orders (Mekhijan, Kumar, Kuehn, Bentley, Teater, & Thomas, 2002) and improved adherence to clinical guidelines (Overhage & Tierney 1997).

However, these systems are not a panacea. In some cases, their use has resulted in negative outcomes, including increased physician order entry time (Bates et al., 1994; Shu et al., 2001) and an increased coordination load on clinical-care teams (Cheng et al., 2003).

For computerized physician-order- entry systems to be fully successful, physicians need to use them. However, as the studies previously discussed indicate, the gains achieved come at a cost, frequently imposed on the physicians using the system. This leads to issues of HCI design, usability, and integration within the care delivery processes to ensure that the burden on these users is minimized. The importance of addressing these issues has also been highlighted in several recent studies showing that the potential for new types of prescribing errors can be created by usability problems in the interface and/or lack of integration of the system with clinical work practices (Koppel, Metlay, Cohen, Abaluck, Localio, Kimmel, & Strom, 2005). In certain care situations, such as critical-care environments, this lack of integration between the clinical processes and patient care needs (i.e., urgency of care) can have dire consequences for patients (Han, Carcillo, Venkataraman, Clark, Watson, & Nguyen, 2005). This emphasizes the importance of identifying and resolving potential usability problems prior to implementation, continually monitoring their use, and incrementally improving their usability in the context of the needs of specific clinical-care environments.

Patient Monitoring Systems

Gardner and Shabot (2001) defined patient monitoring as "repeated or continuous observations or measurement of the patient, his or her physiological function, and the function of life support equipment, for the purpose of guiding management decisions" (p. 444). Electronic patient monitors are used to collect, display, store, and interpret physiological data. Increasingly, such data are collected using newly developed sensors from patients in all care settings as well as in patients' own homes. While such data can be extremely useful for diagnosis, monitoring, alerts, as well as treatment suggestions, the amount, diversity, and complexity of data collected presents challenges to HCI design.

Imaging Systems

Imaging is a central element of the healthcare process for diagnosis, treatment plan design, image-guided treatment, and treatment-response evaluation. Greenes and Brinkley (2001) noted that the proliferation in number and kind of images generated in healthcare led to the creation of a subdiscipline of medical informatics called "imaging informatics." In their review and summary of the field, Greenes and Brinkley noted that:

> as processing power and storage have become less expensive, newer, computationally intensive capabilities have been widely adopted. Widespread access to images and reports will be demanded throughout healthcare delivery networks. . . . We will see significant growth in image-guided surgery and advances in image-guided minimally invasive therapy as imaging is integrated in real time with the treatment process. Telesurgery will be feasible. (pp. 534–536)

They also highlight that such ambitious evolution of imaging systems will be in part dependent on continued advances in user interfaces and software functionality.

Information-Retrieval Systems

Hersh, Detmer, and Frisse (2001) defined information retrieval as the "science and practice of indentification and efficient use of recorded media" (p. 539). In their review of the systems, they addressed four elements of the information-retrieval process: (a) indexing, (b) query formulation, (c) retrieval, and (d) evaluation and refinement. *Indexing* is the process by which content is represented and stored, and *query formulation* is the process by which user information needs are translated in terms of an actionable query. Focusing on clinicians' use, Hersh et al. (2001) pointed out that the evolution of the Internet has posed new and so far unmet challenges regarding the indexing of (essentially loosely and irregularly structured) documents. They call for a unified medical-language system. Thinking of novice users such as consumers and patients, another significant challenge resides in query formulation, since novice users generally have a hard time expressing ill-defined information needs because of their lack of domain knowledge.

Decision-Support Systems

Decision-support systems (DSS) are computer systems used to support complex decision making. The goal of DSS is to improve the efficiency and/or effectiveness of decision-making processes (Shim, Warkentin, Courtney, Power, Sharda, & Carlsson, 2002). As such, many of the complex decision processes in healthcare can benefit from the use of DSS. One class of DSS in healthcare is clinical decision support systems (CDSS), which provided decision support to clinical users during the process of providing patient care. Another class of DSS, patient-decision-support systems (Patient DSS), is designed to educate and support patients as they make decisions that affect their healthcare. The following sections provides a brief overview of these two types of healthcare-related DSS.

Clinical- Decision-Support Systems

Musen, Shahar, Shortliffe (2001) defined a clinical decision support system (DSS) as "any computer program designed to help health professionals make clinical decisions" (p. 575). They characterized clinical decision support systems along five dimensions:

1. system function
2. mode for giving advice
3. style of communication
4. underlying decision-making process
5. HCI

A large number of applications already exists and can be further developed. The opportunities are almost unlimited. Bates et al. (2001) proposed that appropriate increases in the use of information technology in healthcare, especially the introduction of clinical decision-support systems and better linkages in and among systems, could result in substantial reduction in medical errors. For example, studies have demonstrated that implementation of CPOE systems that include CDSS functions (i.e., drug interaction checking, allergy checking, etc.) have reduced medication errors and adverse drug events in both general-care settings (e.g., Bates, Teich, Lee, Seger, Kuperman, Ma'Luf, Boyle, Leape, 1999) and pediatric-care settings (i.e., Potts, Barr, Gregory, Wright, & Patel, 2004; Cordero & Kuehn, 2004).

However, as noted by Musen et al. (2001), "systems can fail . . . if they require that a practitioner interrupt the normal pattern of patient care" (p. 587). Thus, new technologies (mobile devices, wireless networks, and distance-communication technologies) as well as novel HCIs (based on speech, gestures, and virtual reality) offer huge potential to maximize usability and permit seamless integration of clinical decision-support systems within complex, dynamic work processes.

Patient Decision-Support Systems

The medical industry has increasingly acknowledged the need to enable patients to participate in making health-related decisions. In order to effectively participate in decision making, most patients need to become better informed about the options available and need help assimilating that information to apply it to the decision at hand. These health-related decisions range from choosing a health-insurance plan to working with their doctor to select a treatment for cancer.

Patient DSS are one of several types of decision aids used to help patients participate in health-related decisions. *Decision aids* are interventions provided to assist individuals as they make a deliberative choice between two or more alternatives (Bekker, Hewison, & Thornton, 2003). Patient DSS supports a patient in one or more stages of making a health-related decision. Much of the past and current patient DSS research and development efforts have targeted patients with life-threatening or chronic diseases. Most have focused on supporting decisions regarding treatment options (i.e. medical or surgical therapies), although a few have examined early-detection and other issues (O'Connor, 1999). Patient DSS that support patients faced with treatment decisions provide one or more of the following functions that facilitate patient participation in decision making (O'Connor, 1999; Scott & Lenert, 1998):

Educate the patient. Provide the patient with information about the treatment alternatives and outcomes, especially highlighting risks and benefits associated with each treatment alternative.

Tailor information. Tailor information content and/or presentation based on patient characteristics such as their health and demographics factors.

Assess preferences. Use preference-elicitation methods to assess the patient's values/preferences for the possible intervention outcomes.

Optimize decision. Optimize the decision outcome based on context, heuristics, probabilities of outcomes, and algorithms.

Reviews of patient decision aids including patient DSS (i.e., Molenaar, Sprangers, Postma-Schuit, Rutgers, Noorlander, Hedricks, & De Haes, 2000; O'Connor, 1999; O'Connor, Rostom, Fiset, Tetroe, Entwistte, & Llewellyn-Thomas, Homes-Rovner, Barry, Jones 1999; O'Connor, Stacey, Entwistle, Llewellyn-Thomas, Rovner, & Holmes-Rovner, Tait, Tetroe, Fiset, Barry, & Jones 2003b) have revealed that "decision-support strategies have received generally consistent positive ratings by patients in terms of feasibility, acceptability, length, balance, clarity, amount of information, and usefulness in decision making" (O'Connor, 1999, p. 260). Additionally, they demonstrate a number of benefits for patients, including improved decision-relevant knowledge, reduced decisional conflict, improved congruence between values and choice, more realistic expectations, more active patient participation, and fewer patients who are unable to make a decision. Bekker, Hewison, and Thornton (2003) proposed that many observed benefits are a result of decision aids enabling patients to use more effective cognitive and emotional strategies. Note that the reviews do not indicate a consistent, significant effect of decision aids on decision satisfaction, although a generally positive impact on satisfaction with the decision process is indicated. Use of decision aids tends to reduce the number of patients who take a passive role in decision making and the number who are undecided (O'Connor et al., 2003).

In addition to being used to support patients faced with treatment choices, similar support systems can be used to help healthy individuals make health-related choices. For example, DSS can support healthcare consumers as they make the important choice of choosing health insurance. Considerable effort has been made to educate consumers about health insurance. One system available for learning about health insurance plans is the Health Plan Report Card created by the National Committee for Quality Assurance (NCQA) (The National Committee for Quality Assurance, 2003). The report card is, in effect, an interactive decision-support tool that helps consumers compare and select a health plan. The Report Card provides information on a variety of attributes for each plan and enables consumers to identify plans that meet their specifications. This system provides access to plan quality information and uses an easily comparable matrix format and a "star" rating system to simplify comparisons of plans. Despite the increasing availability of information on health plans, many consumers still find the information confusing and the decision difficult. By presenting health-plan information in a way that supports and, perhaps, adapts to a consumer's characteristics and their health-plan preferences, we may improve the usability of the information and enable consumers to make more informed decisions about their health-insurance coverage.

Patient Preferences

Patients' preferences and priorities regarding their health and healthcare are applied each time they participate in making health-related choices. However, an individual's stated preferences can be influenced by external factors. For example, researchers have found that framing of information about treatments is handled (positive, negative, or neutral) has affected patient's treatment preferences (Llewellyn-Thomas, McGreal, & Thiel, 1995). It is likely that these tendencies for external factors to influence individual preferences will hold for a variety of health-related choices. Therefore, presenting decision makers with information previously not considered in their decision may cause them to change their preference structure. Slovic (1995) viewed "preference construction as an active process" (p. 369). Taking this view, it is important for patient DSS designers to take a patient-(user) centered approach to the design of these systems to ensure that patient decision outcomes are improved and to avoid introducing systematic biases into the decision process. By structuring the education and decision process to help the patient develop a comprehensive picture of relevant facets of the decision, we can help the patient develop a more comprehensive, rational set of preferences to apply to his or her health choices. In this way, DSS can help patients/healthcare consumers make more informed decisions.

HCI *Challenges in Patient* DSS

The heterogeneity of patient/healthcare consumer population presents a number of challenges to Patient DSS designers. Patients/consumers vary greatly in age, physical ability, mental ability, computer experience, healthcare/health condition knowledge, and, as discussed previously, decision-related preferences. They also employ difference decision strategies when faced with a multi-attribute decision tasks such as many of those faced in healthcare. These decision strategies influence what information the patient (decision maker) wants to see and how he or she views that information. Decision makers frequently use either compensatory or noncompensatory strategies to evaluate alternatives and make a decision. Compensatory strategies are those in which the information about every alternative is weighed and compared (Johnson, M. M. S., 1990). Comparatively, in noncompensatory strategies, alternatives may be eliminated after an incomplete search, thereby reducing the cognitive load of the decision task (Johnson, M. M. S., 1990). Selection of a decision strategy is related to a number of factors, including age (Johnson, M. M. S., 1990) and personality type. In addition to the characteristics of the individual, selection of a decision strategy is also influenced by decision-task characteristics such as the complexity of the decision (Payne, 1976) and the type of decision aids available (Todd & Benbasat, 2001).

Because of the differing information and decision-support needs of various user populations, designing effective decision support for a broad range of patients/healthcare consumers is challenging. Therefore, it is imperative that Patient DSS designers model the needs of their target user population and design systems to meet those needs. For example, certain patient populations may suffer from limited physical or cognitive abilities, making it vital to create accessible designs that meet their special needs.

EVALUATING COMPUTERIZED HEALTHCARE APPLICATIONS

While there has been a proliferation of web-based healthcare applications available for consumers and patients, the issue of

evaluating applications and guiding users in choosing the best applications has become extremely important. While evaluation is obviously also important for applications targeted at healthcare professionals, the issue is not as critical because healthcare professionals are experts and can exercise their own judgment in the suitability and quality of applications designed to assist them in their work. Consumers and patients, on the other hand, have no or limited basis for exercising such judgment.

Concerned that the growth of the Internet was leading to too much health information with vast chunks of it incomplete, misleading, or inaccurate, Silberg, Lundberg, and Musacchio (1997) proposed four standards for websites:

1. Authorship: authors and contributors, their affiliations, and relevant credentials should be provided.
2. Attribution: references and sources for all content should be listed clearly, and all relevant copyright information noted.
3. Disclosure: website "ownership" should be prominently and fully disclosed, as should any sponsorship, advertising, underwriting, commercial funding arrangements or support, or potential conflicts of interest. This includes arrangements in which links to other sites are posted because of financial considerations. Similar standards should hold in discussion forums.
4. Currency: dates that content was posted and updated should be indicated.

Murray and Rizzolo (1997) pointed out that:

somehow, just the fact that information is traveling quickly through space and being presented on the computer screen lends it an air of authority which may be beyond its due. Sites with official-sounding names can dupe the inexperienced or uncritical into unquestioned acceptance of the content.

In addition to the standards identified by Silberg et al. (1997), Murray and Rizzolo (1997), and others have cited other criteria for evaluating websites. These include the authority of the author/creator, the accuracy of information and comparability with related sources, the workability (user friendliness, connectivity, search access), the purpose of the resource and the nature of the intended users, whether criteria for information inclusion are stated, the scope and comprehensiveness of the materials, and the uniqueness of the resource.

Jadad and Gagliardi (1998) identified a number of instruments used to provide *external* ratings of websites. These ratings are used to produce awards or quality ratings, provide seals of approval, identify sites that are featured as the "best of the web" or "best" in a given category, and/or to declare sites as meeting quality standards. They attempted to determine what criteria were used to establish these ratings and to establish the degree of validation of these rating instruments. However, few organizations listed the criteria behind their ratings and none provided information on interobserver validity or construct validity. Jadad and Gagliardi (1998) also discussed:

- whether it is desirable to evaluate information on the Internet due to concerns over freedom of expression, excessive regulatory control, etc.
- whether it is possible to evaluate information on the Internet due to the lack of a gold standard for quality information and the controversy surrounding its definition.

They also pointed out that information on Internet is "different from that found in journals—information is produced and exchanged by groups of people (i.e., health professionals, consumers, vendors, etc.) using multiple formats (i.e., text, video, sound) modified at fast and unpredictable rate, and linked within a highly elaborate and complex network of Internet sites." They concluded that with respect to *external independent evaluation* of sites it is not clear:

- whether evaluation instruments should exist in the first place,
- whether they measure what they claim to measure,
- whether they lead to more good than harm
- whether users may ever notice, or if they notice, whether they will ignore evidence in support or against desirability, feasibility or benefits of formal evaluations of health information on the Internet.

Robinson et al. (1998) focused more on *internal evaluation* of applications such as self-evaluation by the sponsors or developers of interactive health-communication applications. They pointed out a number of barriers to evaluating these applications. These include the fact that the media and infrastructure underlying applications is in a dynamic state; applications themselves change frequently; many applications are used in situations where a variety of influences on health outcomes exist, few of which are subject to easy assessment or experimental controls; developers lack familiarity with evaluation methods and tools; and developers often believe that evaluation will delay development, increase front-end costs, and have limited impact on sales.

Patrick and Koss (1995) suggested that the effectiveness of consumer health information should be measured by how rapidly and completely desired messages are communicated and how completely intended changes in behavior occur. Saying that Silberg et al.'s criteria (explicit authorship and sponsorship, attribution of sources, and dating of materials) are not enough, Wyatt (1997) provided far more specific direction for evaluating websites. He believes that evaluation of websites should go beyond mere accountability to assess the quality of their content, functions, and likely impact. Evaluating the content should include:

- determining the accuracy of web material by comparing it to the best evidence, i.e., for effectiveness of treatment—randomized trials, for risk factors, cohort studies, or for diagnostic accuracy—blinded comparisons of test with a standard.
- determining timeliness by checking the date on web pages but recognizing that material may not have been current at that time, so need independent comparison with most up to date facts is preferable.
- determining if people can read and understand web material, asking visitors to record satisfaction is unlikely to reveal problems with comprehension since visitors may not realize they misunderstood something or may blame themselves. So, need a minimum reading age for material for the public. However, more accurate to ask users questions based on the web content.

Evaluating the functions of websites should include determining how easy it is to locate a site, how easy it is to locate material within the site and whether the site is actually used and by whom. For those investing resources in a website, with respect to evaluating the impact, he suggests looking at the impact on clinical processes, patient outcomes, and its cost effectiveness compared to other methods of delivering the same information. He recommended using randomized control trials as the most appropriate method for determining impact. Finally, with respect to evaluation methodology, he pointed out the importance of choosing appropriate subjects (not technology enthusiasts) and the need to make reliable and valid measurements. Wyatt (1997) believed that:

> Ideally, investigators would have access to a library of previously validated measurement methods, such as those used for quality of life. However, few methods are available for testing the effect of information resources on doctors and patients, so investigators must usually develop their own and conduct studies to explore their validity and reliability. (p. 1880)

The Science Panel on Interactive Communication and Health (SciPICH) was convened by the Office of Disease Prevention and Health Promotion of the U.S. Department of Health and Human Services to examine interactive health communication technology and its potential impact on the health of the public. The panel was comprised of 14 experts from a variety of disciplines related to interactive technologies and health, including medicine, HCI, public health, communication sciences, educational technology, and health promotion. One of the products of the SciPICH products is an evaluation reporting template (Robinson et al., 1998) for developers and evaluators of interactive health communication applications to help them report evaluation results to those who are considering purchasing or using their applications. The template has four main sections: (a) description of the application, (b) formative and process evaluation, (c) outcome evaluation, and (d) background of evaluators. The panel defined the three different types of evaluation as follows:

Formative evaluation. Used to assess the nature of the problem and the needs of the target audience with a focus on informing and improving program design before implementation. This is conducted prior to or during early application development, and commonly consists of literature reviews, reviews of existing applications, and interviews or focus groups of "experts" or members of the target audience.

Process evaluation. Used to monitor the administrative, organizational, or other operational characteristics of an intervention. This helps developers successfully translate the design into a functional application and is performed during application development. This commonly includes testing the application for functionality and may be known as alpha and beta testing.

Outcome evaluation. Used to examine an intervention's ability to achieve its intended results under ideal conditions (i.e., efficacy) or under real world circumstances (i.e., effectiveness), and its ability to produce benefits in relation to its costs (i.e., efficiency or cost effectiveness). This helps developers learn whether the application is successful at achieving its goals and objectives and is performed after the implementation of the application.

Evaluating the effectiveness of web-based applications designed to relay information and/or enable informed decision making is complicated because the "success" of these particular types of applications is (a) not necessarily always related to observable behaviors; (b) based on the quality *and* usability of the information within the application; and (c) a function of the application itself as well as the users. In terms of outcome-evaluation applications, Robinson et al. (1998) gave examples of the types of questions that such evaluation should address:

1. *How much do users like the application?*
2. *How helpful/useful do users find the application?*
3. *Do users increase their knowledge?*
4. *Do users change their beliefs or attitudes (i.e., self-efficacy, perceived importance, intentions to change behavior, and satisfaction)?*
5. *Do users change their behaviors (i.e., risk factor behaviors, interpersonal interactions, compliance, and use of resources)?*
6. *Are there changes in morbidity or mortality (i.e., symptoms, missed days of school/work, physiologic indicators)?*
7. *Are there effects on cost/resource utilization (i.e., cost-effectiveness analysis)?*
8. *Do organizations or systems change (i.e., resource utilization, and effects on "culture"?).*

However, for websites designed primarily to provide information or enable informed decision making, only the first three questions apply. Other potential outcomes related to change in behavior *might* be applicable depending on the nature of decisions made. Consequently, the evaluation of these types of web guides needs to focus on the use of the guides (assuming that users who "like" an application will use it more than those who do not), the usefulness of the guides, the usability of the guides, the ability of the guide to increase knowledge, and the contribution of the guide to decision making. These elements can be integrated into a conceptual framework as shown in Fig. 33.1 Part of this framework draws from the Agency for Health Care Policy and Research reported on consumer health informatics and patient decision making (1997) as well as Sainfort and Booske (1996).

The framework suggests that to evaluate the effectiveness of web-based health applications, at least three perspectives (at the bottom of the figure) can be taken individually or in combination: (a) the consumer, (b) the site sponsor, and (c) outside experts. The framework posits that the characteristics of the system under evaluation primarily influence accessibility and usability of information. Then, accessibility, in conjunction with consumer/patient characteristics, influence actual *access* to information. In turn, the usability of this information, again in conjunction with consumer characteristics, will influence actual *use* of information by the consumer. Use of information is a complex construct. The framework emphasizes three main uses of information: (a) knowledge (inquiry, verify, learn, augment,

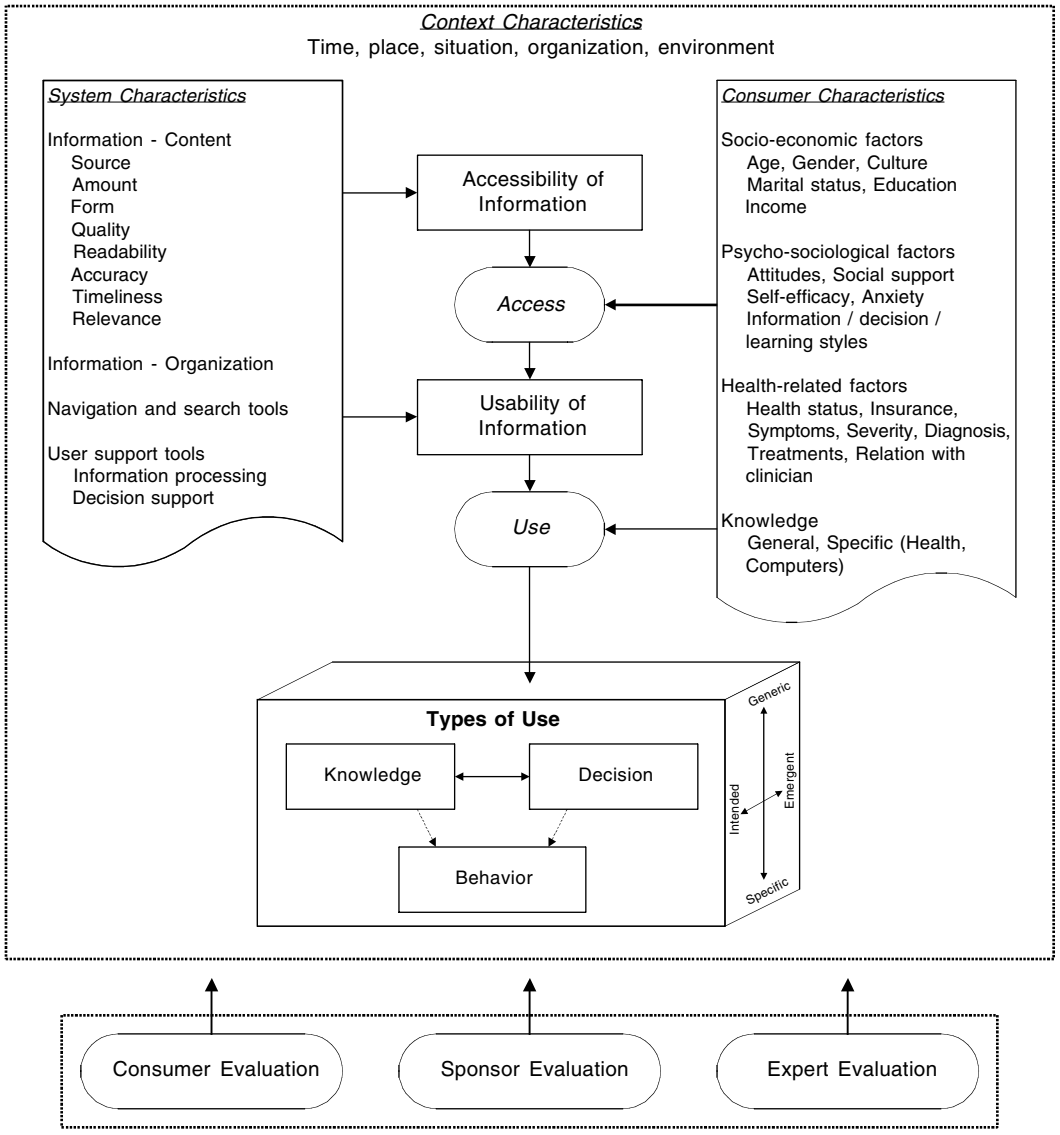


FIGURE 33.1. Conceptual framework.

etc.), (b) decision making, and (c) behavior (intentions and actual behavior change). All three uses are generally interconnected, with behavior usually following knowledge and/or decision making (whether explicitly or implicitly). Also emphasized in the framework is the fact that "use" can be *generic* (common to most web-based health applications) or *specific* (to the application) and also that use of information accessed can be *intended* (i.e., the information was sought to accomplish a specific purpose or use) or *emergent* (i.e., a piece of information accessed triggers a new use). Finally, the framework shows, surrounding this entire process, that the context (the situation, time, place, organization, etc.) can influence all key elements: accessibility, access, usability, and use. Consumer evaluations are formed because of the experiences they have using the system over time. Expert evaluation usually involves assessment of the system itself and its content, as well as an assessment of its (anticipated) impact on users. Sponsors evaluation involves both consumer and expert evaluations as well as consideration of organizational resources expended in the design, development, implementation, and operation of the system.

It is important to differentiate information access and quantity from use and usefulness (Booske & Sainfort, 1998). Indeed, while many studies primarily address attempts to measure the "quantity" of information used, others have acknowledge the need to consider the "quality" of information. The introduction of the web as a source of information has introduced a number of additional considerations in evaluation of the use and usefulness of information. The most common measure of information use on the web is the number of "hits" or "visits" to a particular page. Most of these counts do not differentiate between multiple visits by a single user versus multiple users nor do they consider the amount of time spent on a particular page, i.e.,

whether the page is merely used as a link to elsewhere or as a source of information itself.

In describing three general approaches to information systems evaluation (i.e., the system's output, behavior, and architecture), Orman (1983) defined the quality and quantity of information as the relevant variables in defining an information system's output. However, he went on to say that while both quantity and quality of information were "of considerable theoretical interest but of little practical value since neither can be defined or measured with acceptable precision." Orman defined the quality of information produced by a system in terms of "its contributions to the quality of the decisions it aids" and points out that this is "highly influenced by the style and the behavior of the information user and the state of the environment."

This ties in with the concept of the "value" of information, an important part of traditional decision and economic analysis. Within the field of decision science, the primary use of information is to help *make* decisions. However, an examination of literature from other disciplines has shown that there are other ways to conceptualize the value, and thus the effectiveness of information and its usefulness (Orman, 1983). Although it can be thought that the term "information usefulness" could be an analogue for value of information, within the field of HCI, the concept of usefulness includes additional dimensions. The concepts of "perceived usefulness" and "usability" of information *systems* are both receiving more attention as more systems are developed for novice users. For example, Davis, Bagozzi, and Warshaw (1989) developed scales to measure perceived usefulness and ease of use to evaluate specific software in the work setting. They defined perceived usefulness as "a prospective user's subjective probability that using a specific application system will increase his or her job performance within an organizational context." Ease of use was defined as "the degree to which the prospective user expects the target system to be free of effort." Using a seven-point Likert scale, potential users of software responded to six items addressing perceived usefulness and six to assess perceived ease of use.

Of particular relevance in considering the usability, and potentially the usefulness, of information is the manner of display. Although technological advances continue to increase the number of possible methods for disseminating information, the primary *output* media for healthcare information are still print, video, and sound. Print information is often categorized as text, tables, graphs, and figures. There is a quite extensive body of literature comparing the effectiveness of displaying information in tables versus graphs (Jarvenpaa & Dickson, 1988, and Dahlberg, 1991 provided comprehensive summaries of this empirical work). Many of these studies rely on elicited "directed interpretations" where subjects are given specific questions about the information in tables and graphs. In contrast, a more recent study by Carswell and Ramzy (1997) elicited spontaneous interpretation of a series of tables, bar graphs, and line graphs to find out "what information subjects choose to take away from a display rather than their ability to extract information when promoted."

Within the context of decision making, Schkade and Kleinmuntz (1994) found that different characteristics of information display affected aspects of choice processes. They found that organization of information (such as a matrix versus a list) influenced organization acquisition; form (numeric versus linguistic) influenced information combination and evaluation; while sequence had only a limited effect on acquisition. Johnson, E. J., Payne, and Bettman (1988) and others have provided evidence of preference reversals because of different information displays.

Many of the issues surrounding the use of the Internet are issues of usability. Usability is a central notion of the HCI field. Simply put, Shackel (1991) defined usability as the "the capability to be used by humans easily and effectively" while his more formal operational and goal-oriented definition said, "for a system to be usable, the following must be achieved":

Effectiveness. The required range of tasks must be accomplished better than some required level of performance (i.e., in terms of speed and errors); by some required percentage of the specified target range of users; within some required proportion of the range of usage environments.

Learnability. Within some specified time from commissioning and start of user training; based on some specified amount of user training and support; and within some specified relearning time each time for intermittent users.

Flexibility. With flexibility allowing adaptation to some specified percentage variation in tasks and/or environments beyond those first specified.

Attitude. And within acceptable levels of human cost in terms of tiredness, discomfort, frustration, and personal effort; so that satisfaction causes continued and enhanced usage of the system. (p. 48)

In an attempt to place system usability in relation to other system concepts, Shackel (1991) suggested that:

utility (will it do what is needed functionally?), *usability* (will the users actually work it successfully?) and *likeability* (will the users feel it is suitable?) must be balanced in a trade-off against *cost* (what are the capital and running costs and what are the social and organizational consequences?) to arrive at a decision about *acceptability* (on balance the best possible alternative for purchase), (p. 50).

These concepts are all relevant in a decision to purchase or accept an information system, but may not be all relevant when the resource or product under consideration is information. As Orman (1983) pointed out, the value of an information system is different from the value of its information content, "just as the value of a candy machine is different from the value of the candy it dispenses." Furthermore, usability of information is a necessary but not sufficient criterion for information to be useful. (p. 312)

In the context of informing healthcare consumer decisions, Hibbard and Slovic-Jewett (1997) asked how much information is too much. In response to this question, they determined that the critical element is the ability to interpret and integrate information items: integration is a very difficult cognitive process. They suggest that for a consumer, more information is not necessarily better and that the simple provision is not sufficient when the information is complex. In complex-decision situations, it is important to pay particular attention to human-information processing capabilities and differences

across individuals. For example, specifically addressing information presentation format, Togo and Hood (1992) found a significant interaction effect between gender and format. Personality differences have also been found to contribute to variation in information processing. The way people gather and evaluate information is at least in part based on their psychological type (Slocum & Hellriegel, 1983). Some people are driven to know details before decisions are made while others feel more comfortable assuming what is not known. Individuals vary on how they receive messages, seek information, organize information, and process information. One approach to categorizing cognitive style, the Myers-Briggs Type Indicator (MBTI), is based on Jung's typology (Myers, 1987).

Other models of learning styles can be used to differentiate among individuals and their preferred methods or strategies for taking in and processing information. Felder (1996) provided a useful summary of these models. For example, Kolb's learning-style model classified people as having a preference for (a) *concrete experience* or *abstract conceptualization* (how they take information in) and (b) *active experimentation* or *reflective observation* (how they internalize information). The Hermann Brain Dominance Instrument (HBDI) classified people in terms of their relative preferences for thinking that are based on the task-specialized functioning of the physical brain, i.e., left brain versus right brain, cerebral versus limbic. The Felder-Silverman Learning Style model classified learners along five dimensions: (a) sensing or intuitive, (b) visual or verbal, (c) inductive or deductive, (d) active or reflective, and (e) sequential or global.

In addition to individual differences, other factors that can affect information processing reflect characteristics of the information itself (John & Cole, 1986), i.e., information quantity, information source, information format (mode of presentation, organization, order), information complexity, nature of access (i.e., voluntary vs. mandatory), instructions in the use of the information, response formats (i.e., recognition, recall, judgment, choice), as well as the interface itself.

Understanding cognitive processes involved in accessing, processing, interpreting, and using healthcare information is critical to the successful design and implementation of web-based healthcare applications. This particular point extends to applications targeted at providers and other healthcare professionals. In looking at the impact of computer-based patient record systems on data collection, knowledge organization, and reasoning, Patel, Kushniruk, Yang, and Yale (2000) indicated that exposure to computer-based patient records was associated with changes in physicians' information gathering and reasoning strategies. They concluded that such technology could have profound influence in shaping cognitive behavior. In such systems, the HCI itself can have a strong influence on information gathering, processing, and reasoning strategies. Recently, Patel, Arocha, and Kaufman (2001) surveyed literature on aspects of medical cognition and suggested that:

> cognitive sciences can provide important insights into the nature of the processes involved in human-computer interaction and help improve the design of medical information systems by providing insight into the roles that knowledge, memory, and strategies play in a variety of cognitive activities (p. 324).

FUTURE OPPORTUNITIES AND CHALLENGES

Information and communication technologies have only begun to impact the healthcare industry. Regarding the use of the Internet, a number of applications will be at the leading edge. Mittman and Cain's 1999 prediction that these applications would include consumer health-information services, online support groups for patients and caregivers, healthcare-provider-information services, provider-patient e-mail, communications infrastructure and transaction services, and EMRs appears to true. Similarly, the barriers they predicted would impede or slow down the development of the Internet in healthcare have largely proven true as well: security concerns, weaknesses inherent to web interfaces (especially browsers, search-engine technology, and inability to interact with legacy systems), mixed or lack of quality of information, physician ambivalence, disarray of current healthcare-information systems, lack of resources for web development, and lack of unified standards for electronic communications and transactions.

For information technology to fully impact healthcare delivery with a focus on both patients and providers, these barriers need to be overcome. We believe that the greatest benefits will be reached in the short term by focusing on the following five areas.

Supporting/Enhancing the Patient-Provider Interaction

The Internet will become a critical element of the physician-patient encounter. Organizational websites will evolve from publishing generic consumer content to providing personalized, online services to all consumers (individuals, patients, providers, etc.). The Internet needs to provide for technology that truly supports two-way interaction between patients and providers and directly impacts care delivery.

Effective technologies need to be developed to fully support, enhance, and extend the patient-physician interaction so as to increase the efficiency of care delivery, increase the quality of care received by patients, as well as increase the effectiveness of the work performed by the physician. This latter point is important since a recent study by Linzer, Konrad, Douglas, McMurray, Pathman, Williams, Schwartz, Gerrity, Scheckler, Bigby, Rhodes (2000) showed that time stress, defined as reports by the physician that they needed more time for patients than they were allotted, was significantly related to burnout, job dissatisfaction, and patient-care issues. This study was performed with a national sample (n = 5704) of physicians in primary care specialties and medical and pediatric subspecialties. It demonstrates that the ability to spend sufficient and quality time with each patient is critical to ensure long-term job satisfaction, avoid physician burnout, and increase quality of patient care.

In developing such technologies, as we mentioned earlier, one would need a full understanding of the various cognitive processes involved in information gathering, knowledge acquisition and organization, reasoning strategies, and decision making. Of particular importance is the recognition of various users with various characteristics potentially using technologies in a variety of situations, contexts, organizations, and environments.

Supporting/Enhancing Collaborative Work among Providers

The nature of the physician/associate provider (nurse, physician assistant) relationship is evolving from mere interaction to true collaboration with technologies allowing associate providers to increase involvement in clinical decision making and implementation. Technologies to support collaborative work environment in fast-pace, mobile environment are needed.

As mentioned earlier, the healthcare work environment is very complex and involves a variety of healthcare professionals, all with varying needs for information, knowledge, and support. Designing technologies supporting both individual needs as well as collaboration among individuals presents a number of challenges. In a recent study, Jacko, Sears, and Sorensen (2001) showed that different healthcare professionals (physicians, pharmacists, nurses) exhibit very different patterns of use of the Internet for clinical purposes and have very different perceptions of needed enhancements to support their respective needs. The patterns differed in terms of the range and type of information as well as the depth and specificity of information.

Healthcare organizations will be faced with technology integration challenges and will make use of the Internet as well as intranet technologies to manage and organize the variety of applications in use by providers and their patients. Of critical importance for healthcare organizations will be the design of systems to support clinical decisions, knowledge management, organizational learning, administrative transactions, and supply chain.

Developing and Utilizing New Information and Communication Technologies

Wireless, handheld, and nanotechnologies will drive higher adoption rates of the Internet in clinical settings, patient monitoring, and disease management. Both the number of interactive wireless device users and the number of devices will increase significantly in the next few years. New applications for such technologies need to be developed, based on fundamental research conducted to investigate usage of such devices.

Mobile computing and wireless technologies will increasingly become an important part of healthcare's information technology. Turisco and Case (2001) reported the following predictions:

- The number of wireless Internet users will reach 83 million by the end of 2005.
- By the end of 2004, there will be 95 million browser-enabled cellular phones and more than 13 million web-enabled personal digital assistants.
- The wireless LAN market is expected to reach $1 billion in 2001 and this figure will double by 2004.

Turisco and Case (2001) reported that mobile computing applications for healthcare began with reference tools and then moved to transaction-based systems to automate simple clinical and business tasks. They predict that the next step will be to provide multiple integrated applications on a single device. In particular, they mention the following applications: prescription writing, charge capture and coding, lab order entry and results reporting, clinical documentation, alert messaging, and communication, clinical-decision support, medication administration, and inpatient-care solutions.

Designing and Utilizing Adaptive HCIs

The development and integration of information and communication technologies designed to support, facilitate, and enhance patient/provider interactions are critical. In particular, research is needed to design optimal HCIs. A dual focus on (a) increasing job satisfaction and effectiveness for the medical personnel and (b) increasing quality of care as well as safety for patients is needed. To support these objectives, the HCI must possess at least four qualities:

- multimodal, i.e., have the ability to display and accept information in a combination of visual, aural, and haptic modes,
- personalized, i.e., tailored to respond in a manner best suited to the current user and his or her needs,
- multisensor, i.e., have the ability to detect and transmit changes in the user and/or situation,
- adaptive, i.e., have the ability to change its behavior in real time to accommodate user preferences, user disabilities, and changes in the situation/environment.

Optimal interfaces are critical to virtually all applications connecting people to information technologies and people to people via information and communication technologies. Current research is underway to develop intelligent adaptive multimodal interface systems. In the future, interfaces will automatically adapt themselves to the user (capabilities, disabilities, etc.), task, dialogue, environment, and input/output modes in order to maximize the effectiveness of the HCI.

This is especially critical in healthcare for both types of users: consumers/patients and providers. Consumers/patients present a number of challenges regarding the interface. In addition to presenting different personal and sociodemographic characteristics, users in healthcare present varying degrees of health status: healthy consumers, newly diagnosed patients, chronically ill patients, and/or their caregivers will use the same devices in potentially very different ways. Similarly, different providers will have very different characteristics and will perform a multitude of varying and highly dynamic tasks in different contexts and environments.

Moving to e-Health

The opportunities for improving service and decreasing cost via e-commerce technologies and the supply chain are tremendous. In addition, modern information and communication technologies (including sensors, wireless communication, and implant technologies) will enable electronic delivery of health *care*. This is far more comprehensive than the mere electronic delivery of health information to patients and providers and includes new developments such as telemedicine and virtual reality. Krapichler, Haubner, Losch, Schuhmann, Seeman, and

Englmeierl (1999) claimed that "virtual environments are likely to be used in the daily clinical routine in [the] medicine of tomorrow" (p. 448). However, a number of barriers will need to be overcome. Organizational barriers to e-Health include infrastructure, organization, culture, and strategy, systems integration, and workflow integration. Technological barriers include integration and security, interface design, connectivity, speed, reliability, and usability issues.

The evolution toward e-Health will involve a number of major changes. For example, organizational websites will evolve from publishing generic consumer content to providing personalized, online interactive services to profitable patient segments. Delivery organizations will focus on direct-to-patient relationship building, migrating to a health system truly centered on patients and communities. Communities of interest will rapidly expand to become a force in healthcare navigation. Wireless and handheld technologies will drive higher adoption rates of the Internet in clinical settings. The Internet will become a critical element of the physician-patient encounter and will support and enhance the patient-provider interaction. These changes will lead to a restructuration of the healthcare industry. Digital health plans will emerge and compete or threaten the traditional healthcare-payer business model. Virtual networks will begin to emerge around specialty services to provide efficient personalized healthcare.

References

Adams, W. G., Mann, A. M., & Bauchner, H. (2003). Use of an electronic medical record improves the quality of urban pediatric primary care. *Pediatrics, 111*(3), 626–632.

U.S. Department of Health and Human Services. (1997). *Consumer health informatics and patient decision making* (AHCPR Publication No. 98-N001). Rockville, MD: Author.

Bates, D. W., Boyle, D. L., & Teich, J. M. (1994). *Impact of computerized physician order entry on physician time.* Paper presented at the Annual Symp Comput Appl Med Care.

Bates, D. W., Teich, J. M., Lee, J., Seger, D., Kuperman, G. J., Ma'Luf, N. I., Boyle, D., & Leape, L. (1999). The Impact of Computerized Physician Order Entry on Medication Error Prevention. *J Am Med Inform Assoc,*6(4), 313–21.

Bates, D. W., Cohen, M., Leape, L. L., Overhage, M., Shabot, M. M., & Sheridan, T. (2001). Reducing the frequency of errors in medicine using information technology. *JAMIA, 8*(4), 299–308.

Bekker, H. L., Hewison, J., & Thornton, J. G. (2003). Understanding why decision aids work: Linking process with outcome. *Patient Education and Counseling, 50*, 323–329.

Benton Foundation. (1999). Networking for better care: Health care in the information age. Retrieved December 5, 2001, from http://www.benton.org/Library/health/

BlueCross BlueShield Association. (2006). Medical cost reference guide. Retrieved April 27, 2006, from http://www.bcbs.com/mcrg/chap1/index.html

Booske, B. C., & Sainfort, F: (1998). Relationships between quantitative and qualitative measures of information use. *International Journal of Human-Computer Interaction, 10*(1), 1–21.

Booske, B. C., Sainfort, F., Hundt, A. S., Eliciting Consumer Preferences for Health Plans. *Health Services Research (*1999);*34*(4):839–54.

Carswell, C. M., & Ramzy, C. (1997). Graphing small data sets: Should we bother? *Behaviour and Information Technology, 16*(2), 61–70.

Chassin, M. R., Galvin, R. W., & The National Roundtable on Health Care Quality. (1998). The urgent need to improve health care quality. *JAMIA, 280*(11), 1000–1005.

Cheng, C. H., Goldstein, M. K., Geller, E., & Levitt, R. E. (2003). The Effects of CPOE on ICU workflow: an observational study. In: *AMIA Annual Symposium,* pp. 150–44. November 8–12, 2003., Washington, DC.

Cordero, L., Kuehn, L., Kumar, R. R., Mekhjian H. S. Impact of Computerized Physician Order Entry on Clinical Practice in a Newborn Intensive Care Unit. *Journal of Perinatology* 2004;*24*(2):88–93.

Dahlberg, T. (1991). Effectiveness of report format and aggregation: An approach to matching task characteristics and the nature of formats. *Acta Academie Oeconomicae Helsingiensis Series A:76*. Helsinki, Finland: Helsinki School of Economics and Business Administration.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1999) User acceptance of computer technology: A comparison of two theoretical models. *Management Science, 35*, 982–1003.

Eng, T. R., Maxfield, A., Patrick, K., Deering, M. J., Ratzan, S. C., & Gustafson, D. H. (1998). Access to health information and support: A public highway or a private road? *Journal of the American Medical Association* 1998;*280*(15),1371–1375.

Felder, R. M., "Matters of Style." *ASEE Prism, 6*(4), 18–23 (1996 December).

Ferguson, T. (1997). Health online and the empowered medical consumer. *Journal on Quality Improvement, 23*(5), 251–257.

Fox, S. (2005). *Health information online.* Retrieved April 27, 2006, from http://www.pewinternet.org/pdfs/PIP_Healthtopics_May05.pdf

Gardner, R. M., & Shabot, M. (2001). Patient monitoring systems. In E. H. Shortliffe, L. E. Perreault, G. Wiederhold, & L. Fagan (Eds.), *Medical informatics: Computer applications in health care and biomedicine (2nd ed.)* (pp. 443–484). New York: Springer.

Greenes, R. A., & Brinkley, J. F. (2001). Imaging systems. In E. H. Shortliffe, L. E. Perreault, G. Wiederhold, & L. Fagan (Eds.), *Medical informatics: Computer applications in health care and biomedicine (2nd ed.)* (pp. 485–538). New York: Springer.

Han, Y. Y., Carcillo, J. A., Venkataraman, S. T., Clark, R. S. B., Watson, R. S., Nguyen, T. C., Bayir, & H., Orr, R. A. (2005) Unexpected Increased Mortality After Implementation of a Commercially Sold Computerized Physician Order Entry System. *Pediatrics, 116*(6):1506–12.

Harris, J. (1995). Consumer health information demand and delivery: A preliminary assessment. Partnerships for Networked Health Information for the Public. Rango Mirage, California. May 14–16, 1995. Summary Conference Report. Office of Disease Prevention and Health Promotion, U.S. Department of Health and Human Services, Washington, DC 20201.

Hersh, W. R., Detmer, W. M., & Frisse, E. H. (2001). Information-retrieval systems. In E. H. Shortliffe, L. E. Perreault, G. Wiederhold, & L. Fagan (Eds.), *Medical informatics: Computer applications in health care and biomedicine (2nd ed.)* (pp. 539–572). New York: Springer.

Hibbard, J. H., Slovic, P., & Jewett, J. J. (1997). Informing consumer decisions in health care: Implications from decision-making research. *The Milbank Quarterly, 75*(3), 395–414.

Hoffman, C., Rice, D. P., & Sung, H. Y. (1996). Persons with chronic conditions: Their prevalence and costs. *JAMA, 276*(18), 1473–1479.

Institute of Medicine. (2000). *To err is human: Building a safer health system.* Washington, DC: National Academy Press.

Institute of Medicine. (2001). *Crossing the quality chasm: A new health system for the 21st century.* Washington, DC: National Academy Press.

Jacko, J. A., Sears, A., & Sorensen, S. J. (2001). A framework for usability: Healthcare professionals and the Internet. *Ergonomics, 44*(11), 989–1007.

Jadad, A. R., & Gagliardi, A. (1998). Rating health information on the Internet: Navigating to knowledge or to Babel? *JAMA, 279*(8), 611–614.

Jarvenpaa, S. L., & Dickson, G. W. (1988). Graphics and managerial decision making: Research based guidelines. *Communications of the ACM, 31*(6), 764–774.

John, D. R., & Cole, C. A. (1986). Age differences in information processing: Understanding deficits in young and elder consumers. *Journal of Consumer Research, 13*(12), 297–315.

Johnson, M. M. S. (1990). Age differences in decision making: A process methodology for examining strategic information processing. *Journal of Gerontology, 45*(2), 75–78.

Johnson, E. J., Payne, J. W., & Bettman, J. R. (1988). Information displays and preference reversals. *Organizational Behavior and Human Decision Processes, 42*, 1–21.

Karsh, B., Beasley, J. W., Hagenauer, M. E., & Sainfort, F. (2001). Do electronic medical records improve the quality of medical records? In M. J. Smith & G. Salvendy (Eds.), *Systems, Social and Internationalization Design Aspects of Human-Computer-Interaction* (pp. 908–912). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Kaushal, R., Shojania, K. G., & Bates, D. W. (2003). Effects of Computerized Physician Order Entry and Clinical Decision Support Systems on Medication Safety: A Systematic Review. *Archives of Internal Medicine 163*(12), 1409–1416.

Kennedy, J., & Blum, R. (2001, March 01). HIPAA: The new network security imperative. *Lucent Technologies*, 15.

Koppel, R., Metlay, J. P., Cohen, A., Abaluck, B., Localio, A. R., Kimmel, S. E., & Strom, B. L. (2005). Role of computerized physician order entry systems in facilitating medication errors. *JAMA, 293*(10), 1197–1203.

Krapichler, C., Haubner, M., Losch, A., Schuhmann, D., Seemann, M., & Englmeier, K. H. (1999). Physicians in virtual environments—Multimodal human-computer interaction. *Interacting with Computers, 11*, 427–452.

Kumar, S., & Chandra, C. (2001, March). A healthy change. *IIE Solutions*, 28–33.

Legler, J. D., & Oates, R. (1993). Patients' reactions to physician use of a computerized medical record system during clinical encounters. *The Journal of Family Practice, 37*(3), 241–244.

Levin, I. P., & Jasper, J. D. (1995). Phased narrowing: A new process tracing method for decision making. *Organizational Behavior and Human Decision Processes, 64*(1), 1–8.

The Life and Health Insurance Foundation for Education. (2003). Health insurance. Retrieved November 24, 2003, from http://www.life-line.org/health/index.html

Lindberg, D. A. B., & Humphreys, B. L. (1998). Medicine and health on the Internet: The good, the bad, and the ugly. *JAMA, 280*(15), 1303–1304.

Linzer, M., Konrad, T. R., Douglas, J., McMurray, J. E., Pathman, D. E., Williams, E. S., Schwartz, M. D., Gerrity, M., Scheckler, W., Bigby, J. A., & Rhodes, E. (2000) Managed care, time pressure, and physician job satisfaction: Results from the physician worklife study. *Journal of General Internal Medicine, 15*(7), 441–50.

Llewellyn-Thomas, H. A., McGreal, M. J., & Thiel, E. C. (1995). Cancer patients' decision making and trial-entry preferences: The effects of "framing" information about short-term toxicity and long-term survival. *Medical Decision Making, 15*(1), 4–12.

Mekhjian, H. S., Kumar, R. R., Kuehn, L., Bentley, T. D., Teater, P., Thomas, A., et al. (2002). Immediate benefits realized following implementation of physician order entry at an academic medical center. *JAMIA, 9*(5), 529–539.

Mittman, R. & Cain, M. (1999). *The future of the Internet in health care.* Oakland, CA: California HealthCare Foundation.

Molenaar, S., Sprangers M. A. G., Postma-Schuit F. C. E., Rutgers, E. J. T., Noorlander, J., Hedriks, J., & De Haes, H. (2002). Feasability and effects of decision aids. *Med Decis Making, 20*(1), 112–27.

Murray, P. J., & Rizzolo, M. A. (1997). Reviewing and evaluating websites—some suggested guidelines. *Nursing Standard Online, 11*(45). Retrieved July 30, 1997, from http://www.nursing-standard.co.uk/vol11-45/ol-art.htm

Musen, M. A., Shahar, Y., & Shortliffe, E. H. (2001). Clinical decision-support systems. In E. H. Shortliffe, L. E. Perreault, G. Wiederhold, & L. Fagan (Eds.), *Medical Informatics: Computer Applications in Health Care and Biomedicine* (*2nd ed.*) (pp. 573–609). New York: Springer.

Myers, I. B. (1987). *Introduction to type.* Palo Alto, CA: Consulting Psychologists Press.

The National Committee for Quality Assurance. (2003). *Health plan report card.* Retrieved November 24, 2003, from http://hprc.ncqa.org/index.asp

National Research Council. (2000). *Networking health: prescriptions for the Internet.* Washington DC: National Academy Press.

O'Connor, A. M. (1999). Consumer/patient decision support in the new millennium: Where should our research take us? *Canadian Journal of Nursing Research, 30*(4), 257–261.

O'Connor, A. M., Rostom, A., Fiset, V., Tetroe, J., Entwistle, V., Llewellyn-Thomas, H. A., Homes-Rovner, M., Barry, M., & Jones, J. (1999) Decision aids for patients facing health treatment or screening decisions: Systematic review. *British Medical Journal, 18*, 731–4.

O'Connor, A. M., Stacey, D., Entwistle, V., Llewellyn-Thomas, H. A., Rovner, D., Holmes-Rovner, M., Tait, V., Tetroe, J., Fiset, V., Barry, M., & Jones, J. (2003) The Cochrane database of systematic reviews: Decision aids for people facing health treatment or screening decisions. The Cochrane Library. Available from: http://gateway1.ovid.com:80/ovidweb.cgi.

O'Connor, A. M., Stacey, D., Entwistle, V., Llewellyn-Thomas, H. A., Rovner, D., Holmes-Rovner, M., Tait, V., Tetroe, J., Fiset, V., Barry, M., & Jones, J. (2003) Decision aids for people facing health treatment or screening decisions (Cochrane Methodology Review). In: The Cochrane Library, Issue 4, 2003. Chichester, UK: John Wiley & Sons, Ltd.

Orman, L. (1983). Information independent evaluation of information systems. *Information and Management, 6*, 309–316.

Ornstein, S., & Bearden, A. (1994). Patient perspectives on computer-based medical records. *The Journal of Family Practice, 38*(6), 606–610.

Overhage, J. M., Tierney, W. M., Zhou, X-HA, & McDonald, C. J. (1997). A randomized trial of "corollary orders" to prevent errors of omission. *J Am Med Inform Assoc, 4*(5), 364–75.

Patel, V. L., Kushniruk, A. W., Yang, S., & Yale, J. F. (2000). Impact of a computer-based patient record system on data collection, knowledge organization, and reasoning. *JAMIA, 7*(6), 569–585.

Patel, V. L., Arocha, J. F., & Kaufman, D. R. (2001). A primer on aspects of cognition for medical informatics. *JAMIA, 8*(4), 324–343.

Patrick, K., & Koss, S. (1995). Consumer health information "White Paper." Consumer Health Information Subgroup, Health Information and Application Working Group, Committee on Applications and Technology, Information Infrastructure Task Force. Working Draft, May 15, 1995.

Payne, J. W. (1976). Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance, 16*, 366–87.

Potts, A. L., Barr, F. E., Gregory, D. F., Wright, L., & Patel, N. R. (2004). Computerized physician order entry and medication errors in a pediatric critical care unit. *Pediatrics, 113*(1), 59–63.

Robinson, T. N., Patrick, K., Eng, T. R., & Gustafson, D. (1998). An evidence-based approach to interactive health communication. *JAMA, 280*(14), 1264–1269.

Sainfort, F., & Booske, B. C. (1996). Role of information in consumer selection of health plans. *Health Care Financing Review, 18*(1), 31–54.

Schkade, D. A., & Kleinmuntz, D. N. (1994). Information displays and choice processes: differential effects of organization, form, and sequence. *Organizational Behavior and Human Decision Processes, 57*, 319–337.

Schoen, C., Davis, K., Osborn, R., & Blendon, R. (2000, October). Commonwealth Fund 2000 International Health Policy Survey of Physicians' Perspectives on Quality. New York, NY: Commonwealth Fund.

Scott, G. C., & Lenert, L. A. (1998). Extending contemporary decision support system designs to patient-oriented systems. Paper presented at the American Medical Informatics Association, Orlando, FL.

Shackel, B. (1991). Usability—Context, framework, definition, design and evaluation. In B. Shackel & S. Richardson (Eds.), *Human factors for informatics usability* (pp. 376–80). Cambridge, UK: Cambridge University Press.

Silberg, W. M., Lundberg, G. D., & Musacchio, R. A. (1997). Assessing, controlling, and assuring the quality of medical information on the Internet: Caveant lector et viewor—Let the reader and viewer beware. *JAMA, 277*(15), 1244–1245.

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems, 33*, 111–126.

Shu, K., Boyle, D., Spurr, C. D., Horsky, J., Heiman, H., O'Connor, P., et al. (2001). Comparison of time spent writing orders on paper with computerized physician order entry. *Medinfo, 10*(Pt 2), 1207–1211.

Slocum, J. W., & Hellriegel, D. (1983, July/August). A look at how managers minds work. *Business Horizons,* 58–68.

Slovic, P. (1995). The construction of preference. *American Psychologist 50*(5), 364–371.

Sonnenberg, F. A. (1997, January 27). Health information on the Internet: Opportunities and pitfalls. *Arch Intern Med, 157*, 151–152.

Todd, P., & Benbasat, I. (1993). An experimental investigation on the relationship between decision makers, decision aids, and decision making effort. *INFOR, 31*(2), 80–100.

Todd, P., & Benbasat, I. (2001). An experimental investigation of the impact of computer based decision aids on decision making strategies. *Information Systems Research, 2*(2), 87–115.

Togo, D. F., & Hood, J. N. (1992). Quantitative information presentation and gender: An interaction effect. *The Journal of General Psychology, 119*(2), 161–167.

Turisco, F., & Case, J. (2001). *Wireless and mobile computing.* Oakland, CA: California Healthcare Foundation.

U. S. General Accounting Office (1996). *Consumer Health Informatics: Emerging Issues.* Report to the Chairman, Subcommittee on Human Resources and Intergovernmental Relations, House Committee on Government Reform and Oversight. GAO/AIMD-96-86, July 1996. Washington, DC.

Wager, K. A., Lee, F. W., White, A. W., Ward, D. M., & Ornstein, S. M. (2000). Impact of an electronic medical record system on community-based primary care practices. *Journal of the American Board of Family Practice, 13*, 338–348.

Wagner, E. H., Austin, B. T., & Von Korff, M. (1996). Organizing care for patients with chronic illness. *Milbank Quarterly, 74*(4), 511–542.

Workgroup on Electronic Data Interchange. (2000, July/August). HIPAA: Changing the health care landscape. *Oncology Issues*, 21–23.

World Health Organization. (2000). The World Health Report 2000: Health systems: improving performance. Retrieved April 11, 2002, from http://www.who.int/whr/2000/index.htm

Wyatt, J. C. (1997). Commentary: Measuring quality and impact of the World Wide Web. *British Medical Journal, 314*(7098), 1879–1881.

·34·

WHY WE PLAY: AFFECT AND THE FUN OF GAMES

Designing Emotions for Games, Entertainment Interfaces and Interactive Products

Nicole Lazzaro
XEOD*esign*®, *Inc.*

Without Emotion There is No Game **682**
Forget Usability: Make It Fun 682
Emotions Are for More than Entertainment 683
Requirements for a Player Experience Framework **683**
Emotion's Five Impacts on Play 683
Emotion During Play Helps Gamers 683
Requirements for an Emotion Framework for Games 685
Evaluating Relevant Frameworks for Emotion, Products, and Entertainment 685
Flow the Optimal Experience: Csikszentmihalyi 686
Pleasure from Products: Jordan, Norman, and Boorstin 686
Enter Ekman and Facial Expressions 687
The Player Experience Framework **687**
Why We Play Games: Four Fun Keys Method 687
Fiero from Hard Fun Gameplay **688**
Desire for Fiero Helps Players Master Challenge 688
Fiero Enhances a Player's Sense of Progress 689
Hard Fun Creates Challenge with Strategies and Puzzles 690
Curiosity from Easy Fun Gameplay **690**
Curiosity Retains Attention 690
Chain of Easy Fun Emotions Increase Enjoyment 691
Easy Fun: the Bubble Wrap of Game Design 691
Relaxation from Serious Fun Gameplay **692**
Relaxation Changes How Players Think, Feel, and Behave . . . 692
Chain of Serious Fun Emotions Increase Emotions 693
Serious Fun: Enjoyment Helps Accomplish Real Work 693
Amusement from People Fun Gameplay **694**
Amusement Encourages Social Interaction 694
People Fun: Creates More Emotion When Playing Face to Face 694
Some Emotions Require Two People 0
Players Modify Games to Enhance Emotions During Group Play 696
Emotion from Entertainment Interfaces and Interactive Products **696**
Game-Inspired Affect Makes Product Use More Entertaining 696
Playful Product Attributes Increase Emotions During Use 696
Designing Game Affect for Entertainment Access 697
Playful Interaction and Feedback Facilitates Entertainment Access 697
Creating Emotions in Entertainment Access: Apple's iPod Case Study 697
Creating Emotions in Entertainment Access: Microsoft's XBOX 360Case Study 698
Ready to inhale 698
New Directions and Open Issues **698**
New Play With Smaller, Smarter, More Flexible Devices 698
Open Issues 699
Conclusion **699**
Acknowledgments **700**
References **701**

WITHOUT EMOTION THERE IS NO GAME

Shakespeare designed the emotional space between characters; game developers design the emotional space between players and game.

Researchers have only just begun to explore the role emotion plays in human activities. In designing emotional responses, most aesthetic disciplines treat audiences as consumers of content and pay little attention to designing emotions from interaction or contribution. Whether it is a movie, a bottle of perfume, or a website advertisement, a broadcast model is used to elicit emotions from observers rather than participants. Scientific research is now beginning to show how emotion influences cognition and behavior offering new opportunities to solicit emotion through action. Emotion emerges from and plays a part in most activities, from following a goal to just goofing around (Damasio, 1994; Ekman, 2003; Norman, 2004; Lazzaro, 2004b). Video games lead the way as interactive products that create emotion. More emotional than software and more interactive than films, games manipulate player affect to create poignant experiences. How they do this provides lessons for the design of games, entertainment interfaces, and other interactive products.

Emotion is essential to maintain player focus, make decisions, perform, learn, and enjoy the process of play. Emotion-rich stimuli grab players' attention, such as a swashbuckling adventure in *Sid Meier's Pirates!* The emotions surrounding swordplay increase players' immersion and negative affect or mindset focuses players on applying effort to overcome obstacles. Meanwhile, positive affect from finding pirate treasure improves exploration of alternatives. Exploring options to get the player's sticky ball up on top of a table in *Katamari Damacy* is made easier by the positive affect created as it squashes and picks up cutely rendered bon-bons, toys, and other items. Strong emotional states also allow easier formation and recollection of memories, especially if the user's emotion matches the emotion of the item to be remembered (Ekman, 2003). Special moves in *Top Spin Tennis* that offer an emotional response in an opponent are easier to remember and motivate the search for more. Games are innovators in the design of emotional responses integrated into the activity to accelerate it or provide friction for game goals. Each affective state, each emotion, carves a unique signature into a player's psychology, physiology, and behavioral state to create a player experience. Hoping that graphic realism alone will create emotions is nearly the same as adding more background color to fix a usability issue.

Lessons about game experiences inform entertainment interfaces and other product experiences. Games are not just entertaining; they are self-motivating activities. User-experience designers for all types of products can take advantage of how games create these emotions from participation. To transition from interface design to user experience design, interfaces need to be more than "transparent." In games, the interface that makes everything easy robs the player of the fun. Pushing a button for a car to drive itself is less thrilling than winning the Grand Prix. In software, the interface cannot do it all because it does not have the user's knowledge. Therefore, not only must interfaces get out of a user's way, they should allow the user to express him or herself by motivating and supporting the cognitive and behavioral functions required for use. Designers can take inspiration from games to fashion emotional responses during interaction; for instance, computers can detect and respond to user emotion, such as "being more helpful" when software detects frustration in a user's face or fingertips. Like interchangeable colored lenses, games employ emotion-producing attributes to support human performance: rose-colored here for a mood boost, yellow for sharper vision later on, green night vision with a distance indicator for the dark corners, and black to win style points from others. Current usability methods (increasing efficiency, effectiveness, and satisfaction) mostly remove frustration points; they do not yet include techniques to measure and craft other emotions. To exaggerate, a 100% usable product would be boring once it eliminates all the challenge. Customers strap software onto their boots like a crampon, but it should not do the job so well that it climbs the mountain for them (Lazzaro, 2004a). It is not a productivity tool if one button-click creates the whole spreadsheet. User experiences should focus on making the process of the task not only easier, but also more enjoyable.

Forget Usability: Make It Fun

"I don't want to feel like the game just wasted two hours of my life."
—A *Might and Magic* Player

Wasting time has a whole different meaning to a gamer. Unlike a spreadsheet, the outcome of a game is in the experience of play rather than in the quality of the end result and so is harder to quantify. Traditionally, productivity focuses more on designing a process that creates a better end-product or result. Game design, on the other hand, focuses mostly on designing interactive play that is enjoyable in its own right. Rather than efficiency, game enjoyment requires the pure pleasure of the experience and a fair degree of frustration (Lazzaro, 2004a). Similar to user experiences, player experiences are created when what happens in the game affects the player internally as well as externally. Player experiences are the combination of emotion, thoughts, and other sensations that occur inside and in between players during play. Beyond usability, player experiences design focuses on affect as well as ease of use (Fig. 34.1).

Productivity Software Goals	Game Goal
Task completion	Entertainment
Eliminate errors	Fun to beat obstacles
External reward	Intrinsic reward
Outcome-based rewards	Process is its own reward
Intuitive	New things to learn
Reduce workload	Increase workload
Assumes technology needs to be humanized	Assumes humans need to be challenged

FIGURE 34.1. The goals of productivity and game experiences have several important differences. (Source: Lazzaro & Keeker, 2004).

The interactive entertainment offered by games provides unique opportunities to create emotions in the player and unique challenges for the design professional. A game must be usable enough to play but not so usable as to allow someone to push a button and win. Players crave the illusion of superior control that allows them to accomplish more than others including out-thinking the game designer; but not so much control that they lose their way and do not have a good time.

Emotions Are for More than Entertainment

"Experience is the feeling of what happens."
—Damasio, 1994

Playing games in their discretionary time, gamers mainly play for the emotions the games create. Recent neuropsychology research suggests that two interconnected information-process ing systems continually scan the environment to create a person's experience of the world. A person's *cognitive system* interprets and represents the world internally in order to reason, understand, and interact with it. A person's *affective system* interprets external and internal stimuli relative to goals and needs. This affective system kicks in with an emotional and physiological reaction before a cognitive response is ready. Ideas, thoughts, memories, and knowledge are components of cognition; emotions, moods, sentiments, and other internal sensations comprise a person's affective response. (Damasio, 1994; Norman, 2004) From a cognitive-psychology perspective, cognition understands the world and affect evaluates it (Norman). On a basic level, we bring items with positive affect (sweet-tasting, soothing voices, warm to the touch) closer to us. We push (generally speaking) objects with negative affect (bitter, bloody, sharp, diseased) away from us. In the context of games, the study of affect must also include the discussion of enjoyment.

Player experiences emerge inside the player from the process of interacting with the game. Player-experience design crafts these cognitive and affective responses in conjunction with user behavior. Therefore, the design of player experiences must refine not only the gamer's cognitive response to a system, for example, by reducing complexity; it must also design the gamer's affective response, for example, by inspiring interest or rewarding success in order to increase engagement and support cognitive tasks. For the purposes of game design, affect supports cognitive as well as behavioral tasks, because emotion has a significant effect on enjoyment, attention, memory, learning, and performance.

Emotion and cognition walk hand in hand. According to Ekman, beyond entertainment, emotions are about goals and the things we care about (2003). This makes the creation of emotions ideally suited for game design because most gameplay offers goals with levels and scores to indicate progress. In films, we feel emotions only if we somehow identify with the characters on the screen and vicariously feel their emotions (Boorstin, 1990). This also happens in games, but more central to interactive entertainment is when players feel emotions from what they accomplish and fail at during the game. In productivity, feedback on effectiveness, such as creating a sales presentation, happens after the task of creating it is complete. In games, the success-feedback loop is immediate and built into the process of play.

REQUIREMENTS FOR A PLAYER EXPERIENCE FRAMEWORK

Emotion's Five Impacts on Play

"Emotions prepare us to deal with important events without us having to think about what to do."
—(Ekman, 2003, p. 20)

Games entertain with emotions so players enjoy the ride.

Emotions impact player experiences in five ways. What players like most about games is not the packaging, graphics, or the AI, but the total experience that the game creates for the player. This experience lies in the cognitive, affective, and behavioral changes gamers create for themselves as they play. Emotion generates a big part of the entertainment value, the opportunity for challenge and mastery, the thrill from novelty, the ticket to relaxation, and the opportunity to hang out with friends. Customers buy this ride designed by game designers. These emotions create moving gameplay and make a victory taste sweet Figure 34.2.

Not only does the act of play produce emotions, but also emotions and affect substantially influence the player and how he or she plays. Unlike user experiences, the primary aim of player experiences is to move the player emotionally along with or counter to the game goal. Such techniques heighten the emotional response in the player. Emotions entertain players, focus their attention, help them decide, aid their performance, and assist and motivate learning. Enjoyable emotions from gameplay increase motivation to play further in order to experience more emotions, which makes games self-motivational. Emotions are there from the first click to final volley.



FIGURE 34.2. Game interaction involves and creates many emotions that are clearly visible on the face during play, as seen here during the tutorial for a popular action game. Copyright© 2004 XEODesign, Inc. All rights reserved.

Emotion During Play Helps Gamers

1. *Enjoy:* Creates entertainment from strong shifts in internal sensations
2. *Focus:* Directs effort and attention
3. *Decide:* Aids decision-making
4. *Perform:* Supports different approaches to action and execution
5. *Learn:* Provides motivation for learning, aids in memory and rewards progress

Games heighten emotional responses to increase *enjoyment. Mario Kart*, for example, adjusts speed of players to keep them together for "close" races; likewise, *Jak and Daxter* keeps players just ahead of rolling boulders. A choice between certain death and escape via a narrow window ledge in *Tom Clancy's Splinter Cell* provides *focus* to fill attention in a way that scrolling through options in a word processor's drop-down menu or walking down an office corridor of options does not. Strong exaggerated affect aids player *decisions* whether it is to attack a goblin in *Worlds of Warcraft* (*WOW*) or grant a Sim bathroom privileges in *The Sims*. Gameplay creates moods that aid players to perform; negative affect during a fire fight in *Battlefield* improves pursuing a narrow course of alternatives such as stopping a sniper, while positive affect from munching letter sounds in *Bookworm* improves identification of new word patterns in this *Scrabble*-like tile game. A player *learns* to keep diners happy in *Diner Dash* when an angry customer empties the tip jar.

1. *Enjoy:* Emotions create strong shifts in internal sensations to heighten and refresh player experiences

Suspend players over boiling lava or confront them with hideous boss monsters and the game heightens their emotions. Players look for emotional rewards as well as a high score. Much of the enjoyment comes from the player's affective response and ultimately separates player experiences from user experiences. In games, player participation is essential. The emotions come from the players' efforts in accomplishing a task and game enjoyment centers on the experience of strong emotional shifts from their actions. Games provide the environment where the player becomes the central hero to accomplish the extraordinary. The buddy rescued or the enemy vanquished in *Battlefield* is no more real than in film, but the player's role in the achievement is. An interactive medium, the choices offered by the game (including the graphical user interface) must sharply enhance the experience. A poor interface reduces interactivity and harms the game experience. Therefore, a large part of good game design lies in creating effective interfaces that also create emotional responses. Movies, by comparison, invite the audience to share in the joys and sorrows of characters on screen. Where games move beyond film to claim their true power is by rewarding player action with emotions. Movies can never hand the audience a jet ski for the thrill of stopping the impending global thermonuclear war. As part of their unique value proposition, games have to.

2. *Focus:* Emotion directs effort and attention aids or influences many aspects of cognition by focusing attention, providing immediate feedback and rewards

Emotion supports cognitive tasks by directing, focusing, and holding attention, creating absorbing engagement and at the same time allowing emotion-laden ideas to dominate thought (Clore & Gasper, 2000 as cited in Brave & Nass, 2003). Events are filtered through mood' we attend more to thoughts that match the current mood, and ideas similar to a user's current mood are remembered better (Ekman, 2003; Thorson & Friestaad, 1985 as cited in Brave & Nass, 2003). Emotional stimuli rewards more detailed analysis.

In games, tight feedback loops reward player actions with immediate visual and audio feedback that motivate the player to want to take another action. This rewarding-stimulus-response loop is a powerful motivating force that reaches a wide mass-market audience. Several game-design techniques magnify the effect emotions already have to focus players on a task. For example, game obstacles sometimes use negative affect to increase player focus and, at other times, they use positive affect to increase creativity and problem solving to provide interesting behaviors or situations to encourage players to explore (using positive affect to increase creativity and problem solving). Emotions reward attention. Game feedback provides new stimuli to interpret and creates new experience. Rewards create positive and negative affect. Failures create negative affect with the hopes that players will double their efforts and try again.

3. *Decide:* Emotion is vital to decision-making in games

Humans use both thought and emotion to make decisions. Experimental evidence suggests that people with damage to brain structures involved in emotions can generate appropriate logical options and discuss each decision's impacts and tradeoffs with great clarity, but are unable to make the actual choice itself (Damasio, 1994; Norman, 2004). When we select an entree from a restaurant menu, it is what we "feel" like having. There are also logical components to that choice (calories, fiber, who's paying, etc.), but there is something other than logic that happens inside that helps us choose and "feel" good about the selection. Some people even feel that their car drives better after it has been washed or had the oil changed (positive affect). Emotions also let people make snap decisions (Ekman, 2003; Norman, 2004). Emotions help players decide and combine with logic to make these decisions interesting. Because positive and negative affect guides players, it is easier to choose between options with strong emotional stimuli. Game interfaces that supply strong emotional responses also have this effect. Games also use affect to add conflict. Either the emotions support the objective or make achieving it more difficult as players resist the urge to run away. Recognizing this initial role emotion plays in decision-making offers a crucial way to improve decision-support products.

4. *Perform:* Emotion supports different approaches to action and execution

Emotions are a key component in most tasks. This is easy to see in games, which are often designed to create a particular affective state, which sometimes requires following strict detailed procedures with zero tolerance for error or creative exploration of alternative options. The lesson for human computer interaction (HCI) is that properly designed emotions support the right

affective state to get the job done as well as increase appeal. In productivity software, this is important for both the task (such as struggling to find the right word while writing) and the software (such as struggling to find the right feature), and both create frustration. Like a car's seatbelt alarm, certain levels of frustration create mild negative affect in the user and direct the user toward a certain action. Too much negative affect (such as when players cannot keep cars on the racetrack), makes players feel like they will fail and quit. Software needs aspects that create emotions to support tasks, and allow options that let users balance their own levels of frustration. Like cycles of hard and easy things to do in a game, users need to experience new emotions to refresh their experiences. Emotion creates moods that persist and help the player perform. The emotion the designer chooses should help the actions required. Negative affect narrows attention on aspects relevant to the problem, while positive affect opens it to explore new alternatives. Positive moods influence creativity and flexible thinking for problem solving. Relaxed and happy thought processes expand and become more creative and imaginative, and make players more tolerant of minor difficulties (Norman, 2004). Moods with a negative affect focus attention, while positive moods help players generate new options. Cycling between moods creates variety to refresh the experience, and offers the option of approaching a problem with a different problem-solving strategy related to the new mood.

5. *Learn:* emotion provides motivation and rewards rrogress

Played for pure enjoyment, game emotion motivates players to pay closer attention and repeat an action, enabling them to master highly complex interfaces and interactions, learn countless features and strategies, and spend hours doing this even though they may fail repeatedly. To increase product mastery, many game methods apply nongame interfaces. While productivity software users prefer to learn the bare minimum number of features to accomplish their work rather than achieving level 42 in spreadsheet wizardryness, game-like motivation may increase exploration of additional product features. All games involve learning. Emotions reward changes and growth inside players as they master what they could not do before. Learning and remembering are easier for emotional stimuli, and when experiencing an emotion or a mood, it is easier to remember thoughts that have a matching emotional context. Improved performance such as learning a new skill or high score is a big part of the enjoyment of play.

Requirements for an Emotion Framework for Games

"The problem with the words 'enjoyment' and 'happiness' is that they're not specific enough; they imply a single state of mind and feeling, in the same way that the terms 'upset' and 'negative' don't reveal whether someone is sad, angry, or disgusted."

—(Ekman, 2003 p.190)

A practical methodology for designing emotions should support designers as well as researchers to build better player experiences. More than a model, a practical method for designing and examining player experiences should take into account four perspectives: (a) what players like most, (b) what creates emotions, (c) what game designers can control, and (d) what researchers can measure. The method's components must be observable, salient to the player, relevant to the player's experience of fun, and apply to a wide variety of game genres and hardware platforms. It must account for how activities offered by best-selling games make them more popular than others with similar features. It should track internal and external aspects of play including complex emotions and their impact on players' lives. A framework must capture emotions' role in enjoyment, maintaining player interest, decision making, performance, and learning. Designing for emotion in games should build on the research and theories of several disciplines. It must cover differences in expectations and play patterns such as Bartle who first classified players as (a) Achievers, (b) Explorers, (c) Socializers or (d) Player Killers (Bartle, a). It must measure emotion from play from different game components such as challenge and fantasy (Malone, 1981) and biometrics such as heart rate, control pressure and facial expression (Mandryk, 2004; Sykes & Brown, 2003; Hazlett, 2003) and specific emotional states such as pleasure and aggression using FMRI (Weber, Ritterfeld, & Mathiak 2006). It should cover emotions from game goals; open-ended play; as well as emotions from game events and those that happen in real life. Most importantly, the method must describe which gameplay mechanisms produce specific emotions that are central components of the player experience (Fig. 34.3).

Evaluating Relevant Frameworks for Emotion, Products, and Entertainment

A limitation shared by current models of emotions and products is that they focus more on positive and negative affect than on how to create specific emotions from interaction; and even less attention is given to how the actions of a game player create specific emotions. To be useful to game and product designers, player-experience methodologies must identify more emotions than simply positive and negative affect. Affective states grow and change during gameplay.

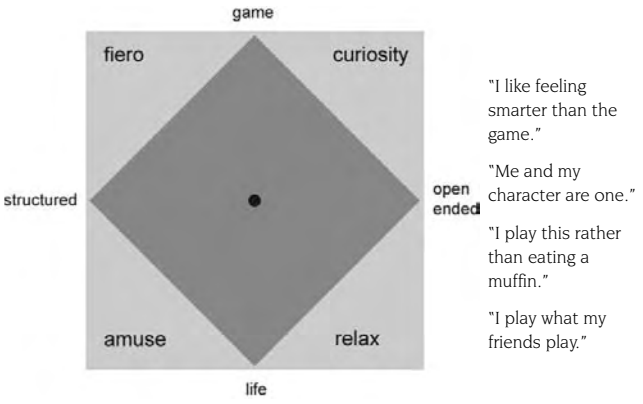


FIGURE 34.3. Players play for "the ride" that games takes them on. Player experiences (PX) generate emotions from reactions along two axes: goal-directed versus open-ended play, what happens inside the game versus how gameplay relates to other aspects of the player's life. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

Many emotions are enjoyable, such as curiosity, which can also lead to other positive emotions such as wonder and even love. However, players also enjoy play experiences containing negative emotions. The pleasure that comes from many game designs requires mastering difficult situations or experiencing unpleasant emotions. How emotions grow and change during play is a critical part of player experiences. During game testing, designers need more information than whether an action was or not fun. Player observation leads to the design of better games, but this requires a deeper understanding of the player experience and better techniques on how to measure it (Fig. 34.4).

Flow the Optimal Experience: Csikszentmihalyi

The most relevant and influential psychological research outside of games is Csikszentmihalyi's model of on optimal experiences or "flow" (Csikszentmihalyi, 1990). He found that people are happiest when engaged in intrinsically motivating activities such as rock climbing, dancing, and gardening. These activities offer long-term, deep, memorable experiences that require concentration and growth. He noted that optimal experiences carefully balance skill with difficulty to create a deeply enjoyable mental and physical state. Without challenge, an activity becomes routine and boredom sets in. With too much challenge, the person becomes too anxious and leaves because he or she feels frustrated (Csikszentmihalyi, 1990). The activity should aim to achieve flow through experience design and create forward movement and personal development through pleasurable engaging experiences. With its balance of difficulty and user skill, flow clearly describes a critical component of play experiences. However, Csikszentmihalyi's model only took into account two emotions: (a) anxiety (frustration) and (b) boredom, and ignored several other emotions essential for gameplay. While these first two emotions

Comparison of Models and Methods to Create Affect from Products and Entertainment Experiences

XEODesign Four Fun Keys Model	**Hard Fun** **Fiero** Challenge Game, Goal	**Easy Fun** **Curiosity** Novelty, Fantasy Game, Open Ended	**Serious Fun** **Relaxation** Real World Purpose Life, Open Ended	**People Fun** **Amusement** Social Life, Goal
Bartle's Original 4 Player Types (1996, 2003a, 2003b)	Achiever Player Killer	Explorer		Socializer Player Killer
Boorstin (1990)		Voyeuristic Eye Visceral Eye		Vicarious Eye
Csikszentmihalyi (1990) Ekman (2003)	Enjoyment, flow	Pleasure, microflow Auto appraisal, memory of emotion, imagination, reflective appraisal		Empathy w/ another, Violation of social norm, talking about emotion, making facial expression of emotion
Hassenzahl et al. (2000) Kim (2000)	Ergonomic quality	Hedonic quality		Community
La Blanc et al. (2004)	Mechanics, Dynamics Aesthetics	Aesthetics		
Malone (1981)	Challenge	Curiosity Fantasy		
Norman (2004)	Behavioral		Reflective Visceral	Reflective
Piaget (1962)	Formal games with rules	Sensory-motor play Pretend play		
Tiger (1992) Jordan (2000)		Physio-pleasure Psycho-pleasure	Ideo-pleasure	Socio-pleasure
Wright et al. (2003)	Spatial-temporal thread	Compositional thread Sensual thread Emotional thread		
Common Drama and Theater Constructs	Character "Motivation", Plot points, Objectives 3-act structure	Setting, Plot, Story, Character, Suspension of disbelief	Catharsis, Music, Set and costume design	Character Dialog Acting

FIGURE 34.4. Several frameworks describe the emotion resulting from entertainment experience or the use of a product. Comparing their similarities and differences provides interesting insight into the basic requirements for entertainment and product emotions. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

are important in games, many other emotions play an important role in player engagement. Csikszentmihalyi grouped together physical, mental, and aesthetic challenges, which prevented discussing emotional effects of art and audio separately from the challenge of gameplay. A model for emotions in games should connect the emotions most important to games to how games create them.

Pleasure from Products: Jordan, Norman, and Boorstin

Pleasure comes from different aspects of experiencing a product. To define what he calls the new human factors, Jordan expanded on anthropologist Tiger's Four Pleasures model to create a framework for thinking about pleasure from products. Jordan discussed (a) the ideo-pleasure, (b) physio-pleasure, (c) psycho-pleasure, and (d) socio-pleasures of a product (or emotions from a product's idea, physical, psychological, and social attributes). Each of these Four Pleasures describes enjoyment from a different perspective including how the use, ownership, or identification with a product produces emotion (Tiger, 1992, as cited in Jordan, 2000; Jordan, 2000). Changing an aspect can strengthen a desired emotional response. Norman built on and simplified Jordan's model to three layers of mental processing: (a) the Visceral (an automatic biological response), (b) the Behavioral (learned actions), and (c) the Reflective (involving thought, self-image and relationship to others) (Norman, 2004). Interestingly, filmmaker Jon Boorstin also proposed three ways that films delight audiences: (a) the Visceral Eye (enjoyment on the biological level such as automatic sensory reactions to explosions or speed), (b) Voyeuristic Eye (enjoyment through seeing events unfold), and (c) the Vicarious Eye (enjoyment through identification with people on screen) (Boorstin, 1990). Unfortunately, none of these frameworks go deeply into what actions create specific emotions.

Enter Ekman and Facial Expressions

Through his research of universal facial gestures, Paul Ekman's Facial Action Coding System (or FACS) and a compilation of others' research, Ekman's work has increased understanding of specific human emotions' cross-cultural boundaries. His fascinating and highly accessible book *Emotions Revealed* described how to identify emotions through facial gestures and the important roles emotions play in our lives (Ekman, 2003) (Fig. 34.5).

Because of its specificity, the FACS coding system for facial gestures offers promise as an emotional measure for games and software, and is the basis for many of XEODesign's observational studies. However, Eckman's work focused on the identification of individuals' experience of specific emotions and stops short of discussing how products or their use can create these emotions. What game designers need is a method that connects specific emotions to player actions in the game.

All of these models lay the groundwork for what is ultimately needed by game designers—a method for producing specific emotions from the interactions players most appreciate during goal oriented and open ended play. To do this the game industry needs an expanded method to create specific captivating emotions from the best-loved types of gameplay. This is a core focus of XEODesign's independent research.

Six Plus One Universal Emotions with Universal Facial Gestures

Emotion	Example
Frustration:	Figuring out how to get character off a roof in *Tom Clancy's Splinter Cell* (and all-too-often created by usability issues that detract from the player experience)
Fear:	Falling into boiling lava, fast-moving projectiles aimed at the player in *Doom*
Surprise:	Using *Myst's* linking books for the first time to transport to a new world.
Sadness:	When the young magician *Aerith*, in *Final Fantasy* VII is murdered
Amusement:	When two Sims get married in *The Sims*, or rolling over and picking up sumo wrestlers in *Katamari Damacy*
Disgust	Becoming a social outcast (social disgust) after losing the dancing challenge in *Sid Meier's Pirates*!
Curiosity:*	Wanting to know what happens by driving the race track the wrong way in *Project Gotham Racing 3*

*Not all researchers (including Ekman) considered curiosity a universal emotion with a unique facial gesture. I include it here as a seventh emotion because of its importance in games and ease of observation.

FIGURE 34.5. Researchers generally agree that there are at least six emotions with universal facial gestures.

THE PLAYER EXPERIENCE FRAMEWORK

Why We Play Games: Four Fun Keys Model

XEODesign conducted independent research to identify four key processes that create emotion in best-selling games. (Figure 34.6) Each experience involves a different emotion to create a different Player Experience Profile. By presenting a goal and breaking it into small achievable steps, games create emotions from *Hard Fun*, where the frustration of the attempt is compensated by the feelings of accomplishment and mastery from overcoming obstacles. Outside of goals, games provide novel opportunities for interaction, exploration, and imagination, which create *Easy Fun*. Games that use emotions in play to motivate real-world benefits to help players change how they think, feel, and behave or to accomplish real work create *Serious Fun*. Finally, games that invite friends along get an interpersonal emotional boost from *People Fun* (Lazzaro, 2004b). The Four Fun Keys are a collection of related game interactions (game mechanics) that deliver what players like most about games. Each offers a key to "unlock" unique emotions such as frustration, curiosity, relaxation, excitement, and amusement. Best-selling games provide features that support at least three of these Four Fun Keys to create a wider emotional response in the player. To

The 4 Fun Keys Model for Emotion through Gameplay

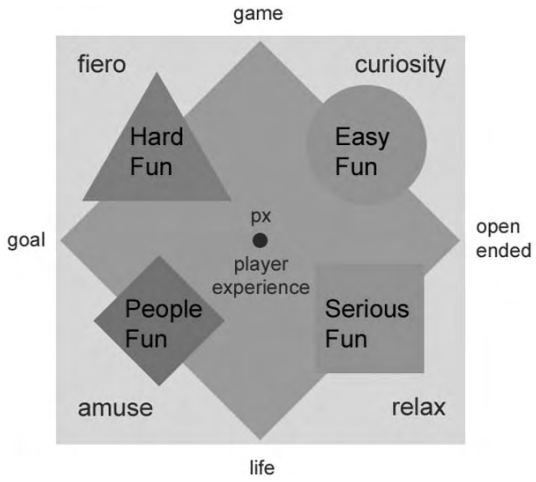


FIGURE 34.6. Each Fun Key is a collection of game mechanics that create a favorite aspect of gameplay. Emotions prepare players for action and reward players. These Four Fun Keys each contribute to a unique set of emotions to the game. Changing one of the four key mechanisms will change the Player Experience profile of the game. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

keep things fresh during a single-play session, gamers move between the four different play styles (Lazzaro, 2004b). Developing each key focuses and rewards the player with emotion from a self-motivating experience that deepens the game's player-experience profile. Designers of products and productivity software can also use these Four Fun Keys to increase emotional engagement for applications outside of games.

Only some of the emotions from playing basketball in the real world come from the Hard Fun of making baskets. Close examination reveals that all four Fun Keys are part of this popular sport. Dribbling the ball or doing tricks like a Harlem Globetrotter offers Easy Fun from novelty and role play. Intentionally blowing away frustration and getting a workout creates Serious Fun. Competition and teamwork make the game even more emotional from People Fun. All four types of fun make basketball's player experience more enjoyable. None of these require story or character. Through examination of how each type of fun creates emotions, designers and researchers can create better and more emotional player experiences (Fig. 34.7).

FIERO FROM HARD FUN GAMEPLAY

Gameplay that rewards effort through challenges to create fiero.

Desire for Fiero Helps Players Master Challenge

"It's easy to tell what games my husband enjoys the most. If he screams 'I hate it. I hate it. I hate it,' then I know two things: a) he will finish it and b) buy version two. If he doesn't say this, he'll put it down after a couple of hours."

—Wife of a hardcore PC gamer

Four Fun Keys to More Emotion through Gameplay

Lead Emotion	Fun Key	Key Game Mechanic
Fiero*	Hard Fun	Affect related to challenge, strategy, and mastery
Curiosity	Easy Fun	Affect related to novelty, ambiguity, detail, fantasy, role-play, and absorbing attention
Relaxation	Serious Fun	Affect related to purposefully changing oneself, learning or doing real work
Amusement	People Fun	Affect related to competition, cooperation, and socializing with others

*Personal triumph over adversity.

FIGURE 34.7. Game mechanics create emotion from what players like most about play. To this group of easily observable emotions, we add frequently reported emotions collected through verbal descriptions of internal sensations as well as player's body language during play. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

The most obvious enjoyment in games comes from mastering a challenge and reaching a goal. Hard Fun is a self-motivating activity that keeps the user focused and enthusiastic by providing an obstacle, an objective, and a score. Hard Fun game mechanics challenge a player to overcome an obstacle to achieve a goal. Hard Fun experiences reward mastery, either explicitly with points and bonuses or implicitly through new levels or abilities. Because this type of play requires application of effort, we call this Hard Fun. In our research, we expand on this phrase (Csikszentmihalyi, 1990; Papert) to define Hard Fun as the rewarding process of mastering a challenge that involves the creation and testing of strategies and the application of effort. Hard Fun rewards effort and discourages failed approaches. Hard Fun creates the emotions of frustration and boredom. More importantly, it produces *fiero*, the Italian word Isabella Poggi and Ekman used to describe the personal feeling of triumph over adversity (Poggi, as cited in Ekman, 2003). One of the most important game emotions, fiero is a strong feeling of personally accomplishing something difficult such as defeating the boss monster (Figure 34.8).

Hard Fun requires a high investment of energy from the player. By perfectly balancing player skill with game difficulty, Hard Fun meets many of the characteristics and requirements for flow (Csikszentmihalyi, 1990; Lazzaro, 2004b). For example, *Pac Man's* simple game mechanic (eat dots and avoid ghosts) offers clear long-term and short-term goals, the opportunity to concentrate, achievable tasks, an uncertain outcome, and immediate feedback to player decisions. It creates a deep sense of control through tight feedback loops between player input and action in the game. All of these enhance players' absorption into a challenging activity and improve their ability to perform. Beyond flow's balance of difficulty and skill, game designers do other things to change how players feel about their progress in the game. According to game designer Steve Meretzky, rewards along the path to the goal enhance enjoyment such as power ups, big jumps in score, animations and sounds (Lazzaro, 2005a). For example, the power ups in *Pac Man* create super-charged feelings as players turn the tables to chase ghosts.

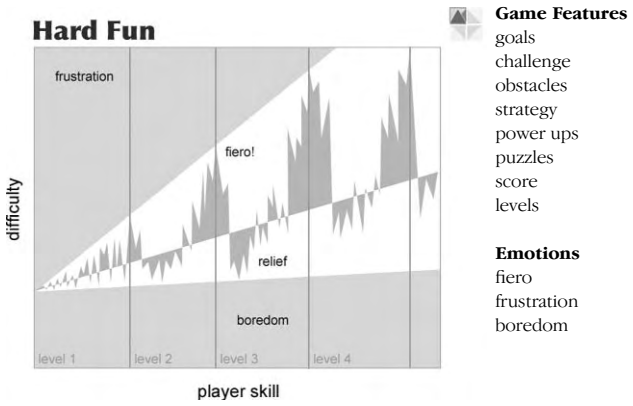


FIGURE 34.8. Similar to Csikszentmihalyi's concept of flow, games must balance difficulty with player skill and provide enough variety of challenge, strategies, and puzzles. If the player gets too frustrated or bored, he or she leaves the game. Charting a player's progress through a game onto Csikszentmihalyi's flow model illustrates that in addition to the requirements for flow, games can be made more emotional from sharp changes in the level of difficulty for game challenges. Game difficulty increases to the end of a level and then resets at the start of the next. When players succeed at the point where they about to quit, they are more likely to experience fiero. In this figure above fiero occurs at the end of level two before continuing to level three of this game (Modified from Csikszentmihalyi, 1990). Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

Hard Fun focuses player attention on achieving results by providing opposition and constraints; such as removing an alien threat in *Halo* or aligning puzzle pieces in *Tetris*. Often games provide a choice of strategy; and games with high-replay ability offer a choice of goals. For example, *The Sims* offers several winning conditions such as the best-looking house, the most friends, or the most money. The new obstacles, constraints, and tradeoffs from different goals suggest multiple strategies and enhance the challenge. These emotions facilitate achievement of a goal, focus play effort, and reward accomplishment. Negative affect increases focus and concentration (Norman, 2004), such as when the player encounters resistance to the goal through failed attempts. Sometimes Hard Fun emotions run counter to what needs to be done. For example, performing well or moving toward something rather than running away is more challenging when surrounded by negative affect. Other Hard Fun emotions, such as fiero from achievement, put the player into a positive affective state by rewarding the player's efforts. Unlike many emotions, fiero does not require an audience (Pioggi as cited in Ekman, 2003) and is a powerful emotion unavailable in film or novels. Players enjoy these emotions from Hard Fun, including the mental focus provided by frustration that helps players concentrate (Fig. 34.9).

While general enjoyment results from emotions that provide motivation, other emotions impact the player's response to the game's stimulus. Decision making is aided or made more challenging by affect, whether it is the positive affect of collecting *BeJeweled*'s colored gemstones or the negative affect from *Bookworm*'s burning letter tiles (which makes their use a priority and picks up the perceived pace of the game). Emotional states also focus effort and attention on available choices in play. Positive affect, for example, helps to generate new ideas such as what gemstones to match in *BeJeweled*; or negative affect, surrounding an impending invasion in *Civilization* provides additional focus for decision making on how to respond.

Fiero Enhances a Player's Sense of Progress

"I have to concentrate!"
—Traveler in St. Louis airport on what he likes most about *JamDat Bowling* on his mobile phone. Playing removes distractions from his consciousness and engrosses him in a rewarding activity while he waits.

Players enjoy many other aspects of games with emotions outside of flow's balance of difficulty and skill. To enhance a player's sense of progress as play continues, games offer new tools or abilities along with new obstacles, constraints, and tradeoffs to maintain interest. Levels divide challenge into gradually increasing amounts of difficulty. The difficulty also increases inside each level, with most levels ending in a "boss monster" final challenge similar to a major plot point in a story. Defeating boss monsters or boss puzzles produces fiero, and then start of the next level is often dramatically easier to provide emotional relief. The best-selling game *Halo* uses Hard Fun to enhance a player's performance by the way it breaks a variety of challenges into levels, rewards progress, provides power ups, such as new weapons or armor, and adds new vehicles to offer fresh strategy options (Fig. 34.10).

Hard Fun Creates Challenge with Strategies and Puzzles

Central to the enjoyment of Hard Fun is the creation and testing of strategies, applying creativity and the development of skills. While some players focus on the goal, many enjoy the process of learning how to win or take pride in how much better they play with each repetition of the game. Unlike real-world sports such as baseball, computer games increase the challenge by changing the rules and winning strategies between levels to keep the emotions from gameplay fresh. Players must strategize to find new ways to play. Rather than requiring more points in less time, best-selling games such as *Collapse* keep players engaged by making a winning strategy obsolete a few levels later (Lazzaro, 2004b). The process of devising new strategies or solving puzzles in new ways is something that players like most about games. The success and failure of these mental tasks and the increase in player skill all create Hard Fun emotion and help players stay engaged and playing.

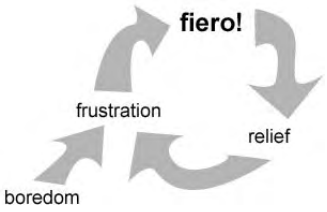
CURIOSITY FROM EASY FUN GAMEPLAY

Curiosity Retains Attention

"Part of the enjoyment comes from the spy technology . . . cool spy tools are part of the Spy experience."
—Xavier playing *Splinter Cell*

Chain of Hard Fun Emotions Increases Enjoyment

Goal-directed gameplay creates Hard Fun emotions that focus and reward players for overcoming obstacles.

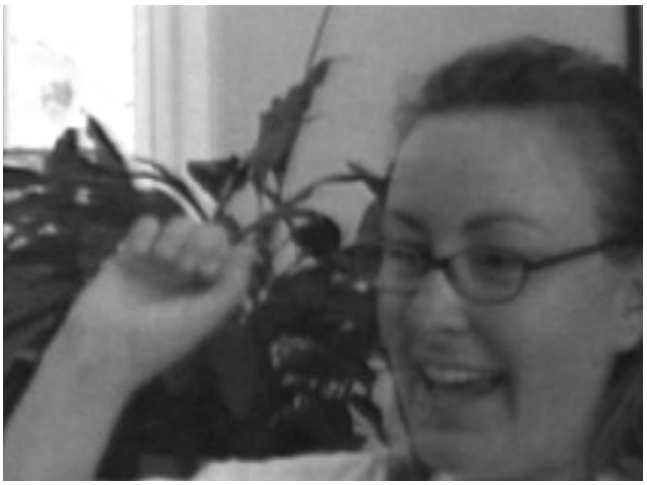


Hard Fun emotions involve obstacles, strategy and success and help players accomplish a goal by focusing and rewarding effort.

Emotion	Common Themes and Triggers
Frustration	Opposition to an important goal, sudden reversal, feeling of being thwarted, physical restraint. Anger prepares the body to remove an obstacle by force. (Ekman, 2003)
Fiero* (Italian)	Personal triumph over adversity. (Ekman, 2003) Overcoming difficult obstacles, players raise their arms over their heads. They do not need to experience anger prior to success, but it does require effort and some frustration.
Boredom**	Repetitive, dull, or tedious tasks. (Ekman, 2003) Lack of interest in the outcome or in playing. Dispelling boredom is also a major reason to play games.

*Fiero is a positive emotion that has a body rather than a facial gesture. It is not yet known what fiero gestures looks like across-cultural boundaries.
**Ekman does not list boredom as having a universal facial gesture; however, it is frequently seen in games that lack sufficient challenge.
(Source: Lazzaro, 2004b)

FIGURE 34.9. Hard Fun Emotion Cycle. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.



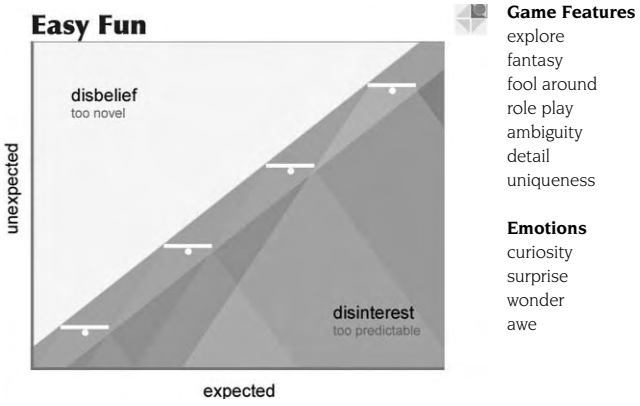
In fiero, the ultimate game emotion, the sensations are powerful. It is how players feel when they beat the boss monster or make level 20 after difficult struggles, or when they win a tennis match at Wimbledon. As fists punch the air a victorious player screams, "Yes! I did it!"

In XEODesign's play lab, fiero appears as a positive upward gesture of the arms or body after succeeding a challenge. Some players jump their characters up and down to express this emotion. (Source: Lazzaro, 2004b)

FIGURE 34.10. In fiero a player raises an arm to show their excitement. Copyright© 2004 XEODesign, Inc. All rights reserved.

Less apparent but equally important to Hard Fun, top-rated games offer a lot of gameplay outside of or en route to a goal. Easy Fun is a self-motivating activity that maintains player engagement through novelty beyond an obstacle, goal, or score. Easy Fun offers novel interaction to inspire player curiosity to explore, fantasize, and role play. By balancing what is expected and unexpected, careful game design prevents the player from quitting from either disinterest or disbelief. Easy Fun derives from the ability to explore and create exceptional experiences unavailable in the real world. The emotions of curiosity, surprise, wonder, and awe surrounding Easy Fun capture and retain the player's attention, as opposed to Hard Fun where the emotions of frustration in hopes of fiero help players focus on and apply effort toward attaining a goal (Fig. 34.11).

If Hard Fun revolves around goal achievement, Easy Fun focuses attention by offering opportunities to explore, get lost in a fantasy, role play, or simply horse around. With compelling Easy Fun, the player ignores the goal completely or forgets about keeping score. Easy Fun players find rewarding, open-ended activities on top of the game's main goal. The so-called sandbox play patterns capture player attention and pull players into deep states of immersion through curiosity instead of daring them with challenge. Best-selling games offer opportunities for interaction through unusual yet enjoyable behavior of the controls and novel interaction with the world. Players become fascinated just by interacting with the game, such as being able to flip a car off freeway exit ramps in *GTA* (*Grand Theft Auto*.) Like the fiero



Gameplay that fills attention through novelty to inspire curiosity.

FIGURE 34.11. Easy Fun emotions maintain player attention without challenge through novelty and inspiring fantasy. Similar to Csikszentmihalyi's concept of flow, players will leave a game because of disbelief or disinterest. To maintain player interest, the game design must balance the expected with the unexpected. The player experience profile of Easy Fun includes curiosity, surprise, and wonder. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

from Hard Fun, Easy Fun has positive peak emotions that reward play such as surprise, wonder, and that "ah-ha" feeling from figuring something out. However, unlike Hard Fun none of these Easy Fun emotions require frustration. Story can often generate the emotions of wonder and surprise, but role-playing and a player's own discovery through exploration create these emotions on a much more personal level. Positive affect encourages creativity and exploration of alternatives (Norman, 2004) that opens the player up to free associations and other possibilities. Easy Fun emotions focus on filling player attention.

Close examination of the conditions, causes, and relationships between emotions provide opportunities for game designers to create even bigger emotional responses in players. It is not that a player cannot feel curious during Hard Fun; but with Easy Fun, curiosity and the sheer joy of interaction drive the player rather than only the score, as it is in Hard Fun. Like improv theater, Easy Fun in games such as *Grand Theft Auto* makes offers to players such as a car and a plate-glass store window. It is up to the player to accept this opportunity and discover how the car and window interact.

Chain of Easy Fun Emotions Increase Enjoyment

Open-ended gameplay creates Easy Fun emotions and increases immersion in an activity (Fig. 34.12).

Easy Fun: the Bubble Wrap of Game Design

"In real life, if a cop pulled me over I'd stop and hand over my driver's license. Here I can run away and see what happens."

—Xavier playing *GTA Vice City*

Easy Fun is the bubble wrap of game design. It is fun without a purpose. Easy Fun provides novel opportunities for interaction that players "discover" alongside of or outside of the main play. The enjoyment and label for Easy Fun comes from the way players goof around, frolic, explore, and play with an ease they do not have when pursuing a specific winning condition as with Hard Fun. Best-selling games such as *GTA, Halo* and *Myst* offer numerous opportunities for Easy Fun; so that when the challenge gets too tough or loses its appeal, the players have many other things to do that create emotional responses. A big role of Easy Fun in best-selling games is to refresh the player between or in the middle of challenges. In *Halo*, for example, players can cycle between the Hard Fun of combat and the Easy Fun of exploring a ring world. The game's battlefield is on the inside of a ring-shaped planet, which inspires curiosity as players investigate as they approach a horizon that instead of dipping down from view, dips up overhead. Easy Fun also provides interest when players pursue the opposite direction of a game goal such as putting the Sims in the pool and removing the ladders in *The Sims* or placing predator and prey animals in the same pen just to see what happens in *Zoo Tycoon* (a flurry of dust and the prey disappears). Through exploring both what's right and what's wrong, games offer more opportunities for emotion, especially from violating norms. In addition to relief from challenge, Easy Fun prevents progress in gameplay from feeling like a skeleton of correct decisions. Easy Fun reinvigorates emotionally and often reinterests the player in the goal. By offering both kinds of fun, the game extends the average play session and lets the player self-regulate the challenge if the Hard Fun becomes too hard.

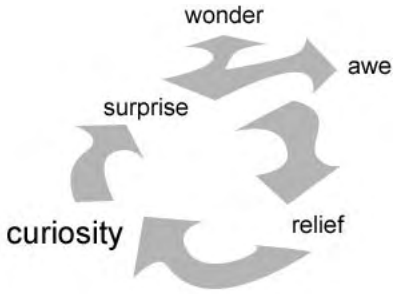
"The journey is the reward."—Design philosophy at Cyan, creators of *Myst*

The emotions from Easy Fun both inspire and satisfy a player's curiosity. To create the emotions of curiosity, surprise, wonder, and awe in addition to novelty, Easy Fun gameplay uses juxtaposition, where contrast between items or events requires the player to investigate. Like Magritte's surrealist painting, "This is not a pipe," a player of *The Sims* must interpret the Siamese pictograph language spoken by the characters. The player projects in and provides an explanation for any conversation between Sims. The Easy Fun of games provides opportunities for fantasy and role play. Players can take elements of the game and do their own thing with them whether it is wielding an orc's mace in *World of Warcraft* (*WOW*) or a donning a superhero's cape in *City of Heros*. Games also reward player curiosity with details such as in Cyan's *Myst*. In addition to the Hard Fun of puzzles, *Myst* offers Easy Fun gameplay from exploring worlds. To encourage players to slow down and notice, *Myst* provides numerous small details in the environment that reward closer inspection and encourage a slow, careful gameplay style.

Instead of running through at top speed, players spend more time exploring and looking for clues to solve the mystery. This supports the style of interaction needed to play and win the game. In addition to detail, *Myst* captures player attention through ambiguity in the setting, surrealistic juxtapositions such as the boiler room inside a tree, linking books and faded sketches of other wondrous technology. The conflict that creates emotional tension in Easy Fun is often between what the

Chain of Easy Fun Emotions Increases Enjoyment

Unstructured sandbox play creates Easy Fun Emotions that reward players outside of challenge and keeping score



Easy Fun offers emotions surrounding the unique. Games provide a sequence of emotions, often starting with curiosity then creating surprise. Easy Fun can sometimes create wonder or awe if the effect is particularly strong.

Emotion	Common Themes and Triggers
Curiosity*	Unusual, unresolved situation that peaks player's inquisitiveness. (Ekman, 2003) Something that players find strange, odd, or intriguing, such as M*yst*'s surreal ship-rock island.
Surprise	Sudden change. Briefest of all emotions, does not feel good or bad, after interpreting the event this emotion merges into fear, relief, etc. (Ekman, 2003) such as when matching two blocks clears the whole board.
Wonder	Overwhelming improbability (Ekman, 2003). Curious items amaze players at their unusualness, unlikelihood and improbability without breaking out of realm of possibilities, such as in M*yst*'s linking books.
Awe	Combination of wonder with a fear and dread (Ekman, 2003), such as a beautiful but impossibly powerful dragon or female warrior in E*verQuest*

*Not all researchers including Ekman recognized curiosity or interest as a universal emotion with a distinct facial gesture. However, those who did recognize it saw the emotion indicated by a lifting and drawing together of the inner eyebrows. In our research, we saw it frequently combined with leaning forward. It was also a feeling reported verbally by players (Lazzaro, 2004b).

FIGURE 34.12. Easy Fun Emotion Cycle. Copyright© 2005–2007 XEODesign, Inc. All rights reserved.

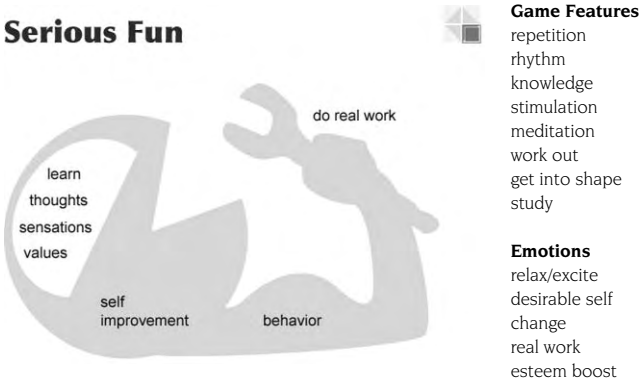
player knows and does not know. Easy Fun offers detail that rewards player exploration and paying closer attention. Because challenges can feel like a grind, Easy Fun refreshes and provides new and interesting things to do in the game. Easy Fun provides novelty to keep play open ended and interesting not because players wonder about whether they have "the stuff" it takes to reach a goal, but simply to make the player experience interesting, surprising, and far from routine.

RELAXATION FROM SERIOUS FUN GAMEPLAY

Relaxation Changes How Players Think, Feel, and Behave

"Playing helps me blow off frustration at my boss."
—a hard core *Halo* player

In Serious Fun, people play games with a purpose (Fig. 34.13). They play to improve their lives by changing how they think, feel,



Gameplay that changes the self and the real world.

FIGURE 34.13. Serious Fun creates emotions to engage in activities that players hope will change how they think, feel, and behave or that will accomplish real work. The player experience profile of Serious Fun includes relaxation, excitement, boosts in self-esteem, and the satisfaction from a job well done. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

or behave. The enjoyment motivates continued engagement with an activity that brings the desired results. The Serious Fun in games reliably relaxes or excites gamers as they play for a purpose to change the player's internal state, develop good habits, improve self-esteem, learn, or do work outside of the game itself. In Serious Fun, the entertainment value captures attention from the emotions surrounding the human values expressed through the act of play. Almost like therapy, players report unparalleled states of concentration and focus during play, making this shift in emotional state one of gaming's biggest takeaways. Whether the activity is fast-action with lots of explosions or slow-paced colored-block matching, players play to experience feelings of excitement, blow away frustration, "get perspective" on their troubles, or create a meditative experience. The stimulation and concentration required drives out bothersome thoughts. Some players simply want to feel more relaxed, excited, or less bored. Players look for a physical and mental workout such as exercise in *Dance Dance Revolutio*n, heightened reflexes in *Project Gotham Racing.* or an increase in mental acuity from word games such as *Bookworm*. Positive and negative affect from play guides the player toward correct moves, and negative affect increases the perceived pace of the game. Real-world benefits include releasing stress therapeutically, learning new vocabulary or math, improving physical fitness through exercise, and increasing a player's mental agility, which some players believe wards off the effects of aging. Physical movement also creates an emotional release felt after exercise and positive feelings from learning and accomplishing tasks in an engaging way. Games played for any of these reasons offer Serious Fun.

Players play to change themselves. Serious Fun is similar to Easy Fun in that players enjoy immersion rather than challenge. However, one aspect of enjoyment many players prefer is achieving a purpose outside the game experience itself such as the ability of the game to calm or excite them. As in Easy Fun, players want the game to fill their attention, but Serious Fun focuses on play's affects long after the game is over. The term Serious Fun captures the real effort and intent that players expend to alter their internal states, express their values and beliefs, and improve their real world skills. Without Serious Fun, a game leaves few long-lasting effects and the play experience feels more like a waste of time. Playing may be hard or easy, but what is important to many players is that the effects of a game last long after the game is over.

Chain of Serious Fun Emotions Increase Emotions

In Serious Fun, many players play to change how they feel, and they take different emotional paths to get there (Fig. 34.14).

Serious Fun: Enjoyment Helps Accomplish Real Work

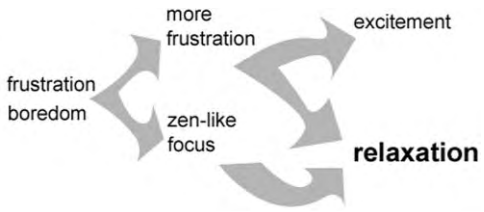
"I felt better about playing [crosswords] because it's good for me. If someone would tell me Tetris *was good for me I'd feel better about playing that."*

—Ellen on doing crosswords. She believes the memory demands of the game will keep her mentally sharp and delay the onset of Alzheimer's disease.

Serious Fun teaches or accomplishes real work. Learning-type games, from multiplication to database administration, uti-

Chain of Serious Fun Emotions Increases Enjoyment

Gameplay that changes has a player feels thinks or behaves creates Serious Fun emotions from creating something players value.



Emotion	Common Themes and Triggers
Excitement	From sudden changes, novelty and challenge (Ekman, 2003). The unexpected catches attention. Many players enjoy the adrenaline rush. Some want to raise their state of arousal, others like how the emotional intensity makes them more relaxed afterwards.
Relaxation	Relief from negative emotion. (Ekman, 2003). Gamers often start a game to attain relief from negative emotions and thoughts prior to playing.

*Self-esteem and knowledge acquisition and the result of exercise are not emotions, but changes in a player's internal states that are reported as desirable results from play (Lazzaro, 2004b).

FIGURE 34.14. Serious Fun Emotion Cycle. A large part of the appeal of games comes from how they change the player inside. This is referred to as Serious Fun. Here a player starts in one of two emotional states and plays a game to end up feeling different. Many gamers appreciate real-world benefits from play. Some play games to become deeply immersed in a process that they hope will change them long after the game is over. Copyright© 2005–2007 XEODesign, Inc. All rights reserved.

lize emotion to encourage the learning of important content or to provide a mental workout, as in an effort to prevent Alzheimer's. Relaxation and excitement make it easier for the player to focus. Serious Fun provides emotions and opportunities for success unavailable to the player in real life by offering a cheaper, safer, more rewarding experience with the content or activity. Rescue simulations can train firefighters about situations that are too dangerous or expensive to do in real time. In educational games such as *DBA Day* (an Oracle workplace simulation XEODesign worked on) players role play a database administrator, including managing their own time. Gee argued that mastering any simulations requires mastering the content embedded in that simulation. By making decisions for the character in a simulation, he believed that simulation games also teach the morals and values of that character coming from their merged identities. The player makes the decisions and the game character has the special abilities to make changes in the virtual world (Gee, 2003). These provide learning opportunities, whether it is values and actions of a thief in *Sly Cooper and the Thievius Raccoonus* or a restaurant owner's time and people-management skills in *Diner Dash* (Lazzaro, 2005b). Serious Fun provides real benefits from play by using game-like structures to reward concentration and focus attention on an activity that is good for players. Leap Frog's *FLY* pen-top computer helps students with their Spanish vocabulary by offering a verbal translation of words written in English; hearing it spoken in another language motivates practice.

Games offer a new spin on learning by capitalizing on player emotion and interaction to create customized training. *Mavis Beacon Teaches Typing* motivates typists with typing games, such as keeping ants out of a picnic basket or driving a racecar. In addition to explicit learning games (such as Oracle's *DBA Day* or an unrelated project, *Doom DBA*), games also embed techniques to offer players more immersive instruction. The introductory experience in *Halo* not only provides a seamless introduction to the user interface by integrating it into the story; but also adjusts the controls to meet the player's preferences without a dialog box. In *Grand Tourismo*, a detailed car-racing simulation game, the opening time trials break down racing skills, such as cornering, into small, easy-to-master steps that fit together to create a more enjoyable practicing experience. By providing a motivating alternative to accomplishing an otherwise boring or unmotivativating task, Serious Fun helps players accomplish real-world objectives such as getting in shape. Someone may play *Dance Dance Revolution* for the excitement of moving to the music, to lose weight by burning calories, or to learn the physical skills and coordination required to dance better. Others play *Karaoke Revolution* to learn how to sing.

Games even make work fun. Players pay for the experience of being a waitress (*Diner Dash*), business owner (*Lemonade Tycoon*), dungeon master (*Dungeon Keeper*), professional sports player (*Madden Football*), or theme-park owner (*Rollercoaster Tycoon*). They even buy the opportunity to sort bugs by color (*Tumblebugs*), pick up their rooms (*Katamari Damacy*), or manage a city (*Sim City*). Games model life problems (*The Sims*) and improve performance during real work. In *The ESP Game*, developed at Carnegie Mellon University, people play a guessing game to enliven the otherwise boring task of providing text labels for images (von Ahn, 2004). Idea or prediction markets use games to beat expert opinion polls. Prediction markets allow participants to express their opinions through buying and selling shares with either virtual or real money. One of the oldest, the University of Iowa's *Iowa Electronic Market*, has allegedly beat expert polls in predicting U.S. presidential elections since 1988. Players of the *Hollywood Stock Exchange* use virtual money to predict what actor, director, or film will receive an Oscar nomination. At *NewsFutures*, players compete for prizes based on their ability to predict news events. Several companies, such as Google, use internal markets to predict launch dates and job openings. Other companies have created internal-prediction markets for sales forecasting (Newsweek, 2004), and managing manufacturing capacity (*Harvard Business Review,* 2003). Not without controversy (gambling is illegal in many countries), markets have even been proposed to predict terrorist attacks (*The New York Times*, 2003). Because playing prediction markets accomplishes a real purpose, it changes how participants feel about playing in general. Like offering a prize for a competition to develop a solar car, having real money or reputation on the line increases the excitement. In Serious Fun, players accomplish real work for many reasons, including to meditate, lose weight, label the every image on Internet, or beat Wall Street predictions.

AMUSEMENT FROM PEOPLE FUN GAMEPLAY

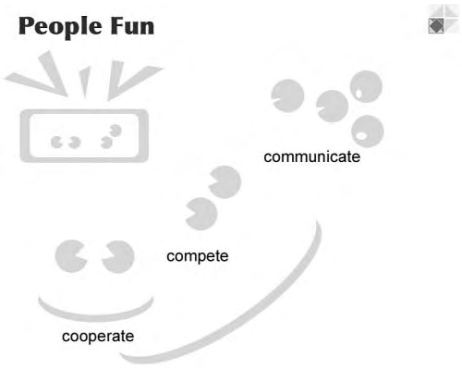
Amusement Encourages Social Interaction

"It's the people that are addictive, not the game."
—Bob, a hardcore sports game player

One advantage that a computer game has is that it is ready to play when you are. No friend required. Still, for many people it is the experience of playing with friends or family that makes play worthwhile. Group gaming provides a mechanism for social interaction, a quick excuse to invite friends over, and more challenges to gameplay. People Fun creates amusement between gamers as they play to spend time with their friends. People report playing games when their friends do, playing games they do not like, and even playing the types of games that they do not like playing games just to spend more time with their friends (Lazzaro, 2004b).

"Since we lost half our guild to Star Wars Galaxies *it's not as fun."*
—A Hard Core Gamer Playing *Dark Age of Camelot*

People Fun is a self-motivating activity that maintains player engagement by supporting interaction with other people (Fig. 34.15). People Fun encourages interaction with other players as they establish social hierarchies, joke, and develop social bonds. It creates affect by providing opportunities to cooperate, compete, and watch others play. Some emotions, such as gratitude, generosity, and schadenfreude, the German word for "pleasure at the misfortune of a rival" (Ekman, 2003), cannot occur while playing alone. When people play together they invent new ways to interact, develop house rules, and add their own content to create more pleasure for themselves and their friends.



Gameplay that involves others to create emotion between players and increase social bonds.

FIGURE 34.15. People Fun creates emotions that increase a player's enjoyment from social interaction. The player experience profile of People Fun includes many emotions that require two people such as Schadenfreude, amusement, naches, and gratitude. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

Nine Pathways to Emotions (Ekman, 2004)	Example from Games
1. Auto appraisal	Explosions, fire, and boiling lava
2. Reflective appraisal	Thinking about last night's dungeon raid
3. Memory of an emotion	Remembering falling from a cliff
4. Imagination	Thinking about what happens from falling off a cliff
5. Talking about	Discussing game events with other players
6. Empathy or witnessing another's emotion	Another player's facial expression or character emote
7. Instruction by others on how to feel	Another player's assessment of an event
8. Violation of social norm	Driving over other players instead of racing against them
9. Making facial expression of an emotion	Smiling and laughing after defeat in front of friends makes it feel more positive

FIGURE 34.16. There are nine ways to create emotions in a person.

The most emotion from games arises when people play together in the same room. Whether playing in a living room or cyber café, the frequency, duration, variety, and intensity of emotional displays all increase (Lazzaro, 2004b). The emotions of schadenfreude, amusement, and naches (Yiddish for the pleasure and pride felt when a child or mentee succeeds) occur more frequently when playing with others. Playing over the Internet, even with video or voice chat, elicits fewer emotional displays than when playing in the same room. A major reason behind this increased emotion is that being in the same room allows for additional interaction between players beyond what is available in the game alone. Players exchange insults and outdo each other with witty commentary. They add content and new rules as real time presence allows for more flexibility (Lazzaro, 2004b). Considered in light of Ekman's nine pathways to creating emotions (Fig. 34.16), People Fun uses more of them than Hard Fun, Easy Fun, or Serious Fun. In fact, the last five out of the nine in Fig. 34.16 are enhanced in group face-to-face play.

People Fun: Creates More Emotion When Playing Face to Face

"Enjoyment when your friends blow up."
—Pat, a PS2 gamer on his favorite game emotion

The involvement of others increases the complexity of the game by creating cooperation and competition, which in turn increases the emotions from play. Different roles for players with shared and opposing objectives increase player interaction. Achieving a goal with a simple rule set becomes exponentially more complex when a player's opponent is another person or group. Not all players like the feelings of rivalry from competition with their friends. Many prefer to team up with their buddies to compete against computer-controlled characters. To encourage group formation, Massively Multiplayer Online Games (MMOs) often have interdependent classes of players, a design that comes from the *Dungeons and Dragons* paper-based role-playing game. In these games, such as *WOW*, character classes encourage the formation of groups, and treasure quests and dangerous dungeons require players to pool their characters' specializations such as archers, swordsmen, hunters, paladins, wizards, and healers. Improved success rates and score bonuses reward players working together in the game and help form the social bonds that increase a player's emotional attachment to the game (Fig. 34.17).

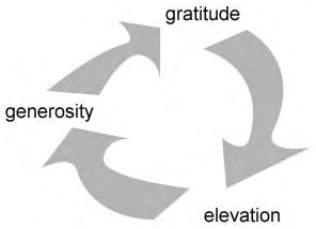
People Fun keeps players going by providing new ways for players to interact. It provides a variety of player-to-player outcomes and bright spectacles for the audience. It provides the opportunity to perform difficult maneuvers and stunts that reward practice with which to amuse and amaze friends. In *Soul Calibur II*, players interact with each other by fighting rather than racing side by side to beat the clock. The frequency and the variety of player-to-player interaction increase commentary between players and make it more enjoyable for an audience to watch. Spectators can participate throughout with comments and criticisms. People Fun games with hidden aspects, special moves, cheat codes, and Easter Eggs (hidden games, objects, rooms, or animations) offer even more opportunity to impress friends. All of these create admiration and envy between players (Lazzaro, 2004b).

Some Emotions Require Two People

People Fun wins out over other types of gameplay with its exclusive access to emotions that require two people, such as schadenfreude and naches. The most frequent emotion in co-located group gaming is amusement. In a group context, even negative events are sources of laughter. The feeling of naches

Chain of People Fun Emotions Increases Enjoyment between Players

People Fun motivates group interaction, interpersonal relationships, and creates emotions between players.



Emotion	Common Themes and Triggers
Amusement	Something that's funny. (Ekman, 2003) Regardless of the emotional tenor of the game, people in groups laugh more than when playing alone.
Schadenfreude*	Pleasure over misfortune of a rival (Ekman, 2003). Competitive players enjoy beating each other, especially a long-term rival. Boasts are made about player prowess and ranking.
Naches*	Pleasure or pride at the accomplishment of a child or mentee (Ekman, 2003). Players enjoy seeing friends succeed as well as the achievements of nonplayer characters they have helped, such as in *Back to Baghdad*, where players can exchange health packs.

*Schadenfreude and naches do not have universal facial gestures; however, they are frequently described by players (Lazzaro, 2004b).

FIGURE 34.17. People Fun Emotion Cycle. In People Fun, some emotions link together like a chain between players, such as offering a healing spell or health pack (generosity) to a fellow player in need (gratitude). A third player feels elevation at witnessing this human kindness and becomes more likely to be generous. Next time a different player may be in need and the links on this chain would rotate. In this way, one game mechanic offers three emotions depending on who starts the chain. Copyright© 2005–2007 XEODesign, Inc. All rights reserved.

and emotional attachment created in *Nintendogs* (a puppy simulation game for the Nintendo DS) are so strong that they not only created a best-selling title, but the title's popularity increased the sales of the hardware platform as well.

People Fun in games is also present in interaction with characters on screen. It enhances emotions with facial expressions for nonplayer characters (NPCs) as well as for player-controlled characters. In *Diner Dash*, players not only have to react to changes in characters' facial expressions but to win the game requires managing customer emotions to prevent them from getting so angry that they leave (Lazzaro, 2005b). Games such as (*World of Warcraft*) increase emotions between players by offering emotes and chat features. Adding voice and video communication between players increases emotions, and co-located group play takes this one step further by supporting face-to-face collaboration in addition to face-to-screen.

Players Modify Games to Enhance Emotions During Group Play

In People Fun, players like playing together so much that they create house rules and convert single player games to make group play more enjoyable. In order to play a *Buzz Lightyear* game together, one family assigns each person a different key to run, jump, or shoot. To share in a pirate adventure, one set of three college students plays *Sid Myer's Pirates!* game by taking turns. In a bike-racing game where players outnumbered the controllers, the winner is the player who "ran the gauntlet" and beat all challengers in the room, not by winning the race, as that would take too long, but by outdistancing each opponent. This said, some players eschew contact with other people if they mostly care about developing strategy or understanding how to play better. To them, playing a game such as *Hearts* online is better when played alone because they do not want to chat or posture with players that they do not know. Such players prefer the Hard Fun even in multiplayer games. Whether it is *WOW* or *Spades*, and depending on their mood, some end up soloing rather than joining a group.

EMOTION FROM ENTERTAINMENT INTERFACES AND INTERACTIVE PRODUCTS

The Four Fun Keys: Hard Fun, Easy Fun, Serious Fun, and People Fun play an important role in making games self-motivating. They provide the opportunity for challenge and mastery, prospect for exploration and discovery, give a boost to self-esteem and an excuse to spend time with friends. The gameplay in games capitalizes on the interaction between human emotion, thought, and behavior. From this perspective, productivity applications have the same goal. Productivity applications take for granted that the user provides the motivation for using the tool, such as workplace responsibilities or a boss's deadline. The compelling and enjoyable nature of computer games makes the process itself enjoyable, so game designers spend a

great deal of design effort to ensure that their games provide a rich emotional experience during play. Study of these interactions offers clues as to how to make productivity tools more motivating without making it a game. The importance of emotions in design is that emotions can influence when, how, and why to use a product.

Game-Inspired Affect Makes Product Use More Entertaining

Games offer many lessons for product-interaction design outside of entertainment, especially in the relationship between design and emotion. Software should make tasks easy by requiring just a few steps and providing appropriate emotional stimuli to focus and reward the user for completing them. One of the biggest differences between productivity software and software used for entertainment is that the pleasure and motivation comes from accomplishing outside tasks, such as completing a spreadsheet or creating a database. Emotions in nongames should create enjoyment, aid decision making, focus effort and attention, and provide motivation. Like games, it is possible to design interactions that are similar to all four types of fun to create a wide spectrum of emotions in nongame software. (Fig. 34.18)

Playful Product Attributes Increase Emotions During Use

Many products employ novel opening techniques to increase customer appeal. Beyond "out-of-the-box" experiences, the experience of the product in use is becoming an important part of a product's competitive advantage. Similar to the way packaging frames consumer expectations and emotional associations with a product, how a product opens or switches on can create emotions and other associations every time the product is used. For example, Danger's *Sidekick* cell-phone lid spins open horizontally with a snap. The *Robo-Book* Light surprises by unfolding like a mechanical arm. The nozzle on *All* liquid laundry soap dispenser features a bright red rubber ball to squeeze for detergent. The designers at IDEO transformed the tiered old carpet sweeper to create the *Swiffer Carpet Flick*. The *Carpet Flick* combines a free-flowing universal joint at the head (Easy Fun), a transparent body so customers miss none of the action (Hard Fun's feedback on goals), and accomplishes real work in an enjoyable way (Serious Fun). The buttons on some productivity software offer Easy Fun if they glow or highlight when moused over. A hard disk defragmenting utility displays progress with an animation that is mesmerizing and game-like to watch, as the colored segments are organized and grouped together. None of these products are games, yet all of them offer Easy Fun with novelty to inspire curiosity, surprise, and a little wonder. Regardless of whether it is a productivity application or a trash can, if it creates enjoyable emotions, it will be used more often and shown off to friends (People Fun).

Creating Affect Through Actions For Nongames

Hard Fun	**Easy Fun**
Affect to increase focus on and enjoyment of a task:	Affect to capture attention and enjoyment:
Spreadsheets	Data mining
Word processing	Searching
Time management	Multivariant problem solving
Decision support	Creativity support
People Fun	**Serious Fun**
Affect to facilitate the interaction and cohesion between individuals:	Affect to help user to accomplish a lot of work consistently over time:
Cell phones	Learning tools
E-mail	Therapies
Human-resource and project-management applications	Medical devices
Groupware	Cleaning products
Presentation software	

FIGURE 34.18. The affect from the Four Fun Keys can increase emotions and enjoyment for several kinds of productivity-software applications and consumer products. Copyright© 2004–2007 XEODesign, Inc. All rights reserved.

Breath-relaxation techniques are more effective if practiced daily. To encourage frequent use with Serious Fun the *Stresseraser* (a home-therapy device that provides bio feedback on the user's pulse and breathing exercises) uses both Hard Fun and Serious Fun. The device rewards customers with a point system and game-like chart feedback of their pulse data. The use of a display graph and a gradual increase in number of points earned per session encourages use by offering Hard Fun and Serious Fun at the same time.

"Why not challenge yourself to see how good you can make yourself feel by getting 100 or more points a day for two months?"

"A session of 30 or more points is a great way to start the day off right. Especially on days when you feel like you got up on the wrong side of the bed."

—(www.stresseraser.com)

The catalog copy on the company's website frames customer expectations in terms of Hard Fun. The game-like intent of the product designers is even more evident in unreleased game modules where customers float a bird around obstacles as a reward for reducing their pulse (Fabricant, 2005). Their approach to relaxation is not about building frustration to eventually succeed through their skill to feel fiero. Instead, with the *Stresseraser*, as in the Interaction Institute's *Brain Ball* game (Hjelm, 2003), people win by relaxing and taking deep breaths. For these products goals and feedback allows players to monitor and change their own emotional state for Hard Fun and Serious Fun.

Designing Game Affect for Entertainment Access

Playful Interaction and Feedback Facilitates Entertainment Access

Game-inspired affect increases the appeal of entertainment applications such as televisions, game consoles, e-zines, and music

players. Because the primary customer goal is to locate content, accessing entertainment options need not be a game in itself. However, entertainment devices can have a game-like feel. They should be even easier to use than productivity applications, because they don't borrow motivation from completing an outside task. They should also create excitement about the entertainment options they provide.

Since the beginning, graphic-user interfaces have employed metaphors such as windows, trashcans, and elevators to explain how features work. Apple's Macintosh OSX operating system established a new trend in modern interface design by going beyond the clear explanation of function to creating an experience. Mac OSX is an interface that is "fun" to use. The dock magnifies icons under the cursor to make them easier to select, at the same time creating pleasurable animation that is pure Easy Fun, even without the goal of opening an application. The genie-out-of-a-bottle animation of document windows as they expand out of a tiny icon on the dock offers more Easy Fun to create curiosity, surprise, and wonder. Operating these features entertains all by its self.

Creating Emotions in Entertainment Access: Apple's iPod Case Study

The most game-inspired device to come out of Apple is the iPod. The iPod's novel combination of hardware and software interfaces creates a new experience full of affect. The novel interaction facilitates entertainment access by offering a customer experience unlike any other. While not technically a game, the iPod uses Hard Fun, Easy Fun, Serious Fun, and People Fun to create emotional experiences for the user.

Central to the user experience of the iPod is the control wheel. It attracts attention and creates emotions of Easy Fun such as curiosity and surprise. Its novel round motion easily scrolls through long play lists. Emotions arising from exploring the free-flowing motion focus attention on finding a song. The quick feedback (including a separate speaker just for the scroll clicks) enhances the sense of mastery (Hard Fun) as a customer gains control. It feels like a game. Even without pressing play, scrolling is enjoyable because using the four directional buttons and wheel mimic a game controller. In addition to the form resembling a stereo speaker, the circular DJ Scratching action takes advantage of an action already associated with making music more exciting and even offers an opportunity to role play for more Easy Fun. The music visualizations on the companion iTunes software offer fascinating animations in time with the music.

A game of *Bricks* (similar to *Breakout*) shipped as an Easter egg (hidden game) in the first edition and later more elaborate games such as Zuma under their own games tab further established the iPod's connection to fun and enjoyment. As easy as the iPod is to use, users must experiment and might experience frustration in learning to understand the menu hierarchy. This offers opportunities for Hard Fun as customers search for their favorite track. Users may experience a feeling of accomplishment rather than fiero when they succeed, but the dancing black silhouette models in the iPod's ad campaigns frolic with arms overhead clearly evokes fiero and other kinds of joy. That the iPod offers Serious Fun of mood-altering music and videos is a given. For additional Serious Fun, iPods also play business-audio books and offer file storage to do the real work of transporting documents and spreadsheets along with music files. The trademark white headphones offer emotions from People Fun even while separating from others auditorally by sending a clear message about the fun-loving social group an to which iPod customer belongs. These status-symbol icons create People Fun emotions from association with belonging to the "in crowd" as well as the emotions expressed in the ads. Several iPod accessories allow customers to share their music during playback. And iTunes' trading of play lists, creation of custom CDs, and podcasting all provide ways for iPod users to reach out to other people for more People Fun.

Creating Emotions in Entertainment Access: Microsoft's XBOX 360 Case Study

Ready to Inhale. The designers of the XBOX 360 video game platform had a specific customer experience in mind. Whereas the design inspiration for the first XBOX was like the Hulk bulging with muscles to break out of a box, the XBOX 360 was to be more of an inhale, like Bruce Lee drawing breath in preparation for play, poised, and ready to strike. Instead of a file-and-folder metaphor, the main UI for the video-game console is designed to capture a moment of preparation before the challenge. The idea was to avoid fussy animations, yet provide something interesting and intuitive to get the player emotionally ready and mentally focused before play begins. The Easy Fun of navigating the menus increases immersion. The emotions for this interface match the player's goals during this part of the use cycle.

In terms of Easy Fun, the menus for the XBOX 360's main UI curve sideways like rib bones and mimic the gesture of inhalation as they move. The menus slip from side to side in a breathing-like motion. The sounds are organic and whip-like with a satisfying thump at the end as the menu snaps into place. This novel auditory and visual experience makes the menus enjoyable to explore without becoming too complicated or intrusive. Novelty of the sideways menus creates curiosity and encourages exploration. The designers were going for playful (the attitude in games) without being so entertaining as to be a game in itself. It is about providing very simple access and a pleasurable experience that creates excitement but doesn't get in the way.

The most important way the XBOX 360 increases enjoyment of games played on its console is through the use of People Fun. To deliver more People Fun emotions and to foster connections between players, the XBOX 360 offers community features for all its titles. Gamers can create their own persistent player profile (or gamer card) that spans game titles. Personal profiles provide an in-game identity and list in-game accomplishments so friends can compare game-specific achievements, scores, and what game they are currently playing. For the nongamers in the household, the XBOX 360 allows access and display of family photos and video for a different kind of social interaction.

The emotions in these user experiences for iPods and XBOX 360s not only help customers accomplish tasks. The UIs create experiences that make the devices pleasurable to use, put users

in the appropriate emotional and cognitive mindset, and capture the next level of the four pleasures that Jordan considers essential in the new human factors: (a) Physio-pleasure, (b) Psycho-pleasure, (c) Ideo-pleasure, and (d) Socio-pleasure (Jordan, 2000). As the focus for HCI shifts from interface design to user experience, more accurate taxonomies for internal experiences and methods to measure these experiences are needed. The development of these tools will facilitate the design of more enjoyable player experiences. While entertainment interfaces such as the iPod and the XBOX 360 have done much to create powerful user experiences on several levels, there is still much more that is possible to do.

NEW DIRECTIONS AND OPEN ISSUES

New Play With Smaller, Smarter, More Flexible Devices

As technology shrinks and integrates itself into more aspects of our lives, people play more games in more places. Devices that are smaller, smarter, mobile, and contextually aware create new opportunities for electronic gaming, which until now has largely been restricted to players at home around a single screen. What happens when the services offered by a laptop become cheap enough to print on a candy wrapper? Mobile and alternate-reality gaming offer the opportunity to make things happen in the real world, which further enhances other experiences such as enjoying the company of friends and dispelling boredom while waiting in line. Games break out into the real world through geo-caching and games of tag through a city. Promotional Alternate Reality Games (ARGs) such as the *The Beast* for the movie *AI* and *I Love Bees* for the game *Halo 2*, use real-world-web-enhanced gameplay to market products and create communities of millions of players (4ourty2wo, 2005). With ubiquitous computing, everything from mobile phones, to the front door, to a ketchup bottle in a diner could contain enough smarts to offer services. Will they all contain a screen and therefore the potential to host a game? Will we surf the net from our saltshaker or will it provide other opportunities to engage our attention?

Many designers chase faster hardware and better graphics, yet the emotional power of games does not occur on screen; it takes place inside the player's head. The biggest emotions that new technology will create is not through rendering blood, sweat, and tears in molecular detail. The real changes in emotional gaming will happen through supporting a more agile design process and by making gaming devices smaller, sharable, context-aware, and more accessible, such as game controllers with fewer than 12 buttons like Nintendo's *Wii*. Until we all get personal holodecks and natural language processing, the success of many new kinds of games will come from connecting the cheapest emotion generators around: a player's rivals and friends.

Life is becoming more game like as games and elements of play inspire new dimensions of product appeal. Like games, products create experiences. Cutting-edge product designers now design customer experiences, not products, as witnessed by the iPod. Designers aim for engagement in addition to making something better/faster/cheaper and easier to use or market. Adding playful elements to goods and services increases the attraction of everything from advertising messages, to Southwest Airline's in-flight safety announcements to the design of public spaces. We see the increasing importance of emotions in design of products such as the playful squid shape of the Phillip Stark orange juicer, the surprising opening rotation of Danger's *Sidekick* mobile phone, and in the pleasing octave chords produced by *Segway's* acoustically designed motors. The progress being made toward this shift toward more emotional design is even more apparent in games.

Games are already redesigning how we work and shop. Employers use games to screen potential hires for three-dimensional (3D) reasoning skills and train them to solve problems with multiple variables. Games changed consumer processes such as buying and selling on eBay or the dining experience at Dave and Busters (Chuck E. Cheese for adults). Even web software applications like Flikr (www.flikr.com) include more fun to increase appeal by sharing items between friends or by offering features to be "gamed," such as displaying the number of friends a person has in *Friendster* or *LinkedIn* (www.friendster.com.www.linkedin.com) social-networking software. Understanding how games create emotions makes products and services more engaging and enjoyable and even improves community and the quality of life.

Open Issues

Better design of the emotions required for and that result from interaction will unlock more human potential from using products and games by priming and rewarding users with appropriate affective states. We know that cognition and emotion walk hand in hand. We are only just beginning to discover how they support each other and how interaction creates emotion. The Four Fun Keys model starts this journey to build and examine player experiences in terms of specific emotions such as fiero from Hard Fun, curiosity from Easy Fun, relaxation from Serious Fun, and amusement from People Fun. Further measurement of specific emotions during different types of cognitive, behavioral, and emotional experiences will help piece together the role individual emotions play in cognition and activities. For example, a surprising event orients an individual's attention to determine whether the source is a benefit or a threat. Surprise does this in addition to creating internal sensations, producing facial expressions, and communicating information about the source to others. Curiosity, the lead emotion from Easy Fun, has a strong cognitive component, which also focuses attention in a pleasurable way. Satisfying curiosity especially when it results in feelings of surprise and wonder produces strong pleasurable sensations that motivate an individual to repeat the activity. Additional research will tell us how to create designs that inspire, maintain, and intensify feelings of curiosity. This is important for applications such as *Google* to improve searching the Internet or when browsing an e-commerce catalog (www.google.com). Conversely, intensifying fiero from finding the search item also improves the user experience. Informed with a deep understanding of how curiosity and frustration employ different affective states to focus attention the design of entertainment and productivity products can achieve the next level of engagement by sculpting emotional responses that compliment the tasks.

CONCLUSION

Games have the creative freedom to push technical boundaries and be light years ahead of productivity interfaces. They have been champions and early adopters of new interface techniques from joysticks to voice commands. Game interfaces lasso new hardware and experiment with dialogs and menus to deliver novel experiences with a vigor never seen in productivity applications. Mastering these innovative interaction techniques provides richer experiences that inspire a devotion to learning features. Radial menus in *The Sims*, audio menus, and draw-your-own-game interfaces in LeapFrog's *FLY* pen-top computer, the Line Rider web game, the iPod's scroll wheel and XBOX 360's organic side-scrolling tabs are all examples of original alternatives to navigating file-and-folder hierarchies and dialog boxes. Games boast heads-up displays and voice commands to allow players to multitask (*Tom Clancy's Rainbow Six* on the *XBOX*) or sing (*Karaoke Revolution*). Still other games use camera interfaces that track the body's movement (*EyeToy* games, *Virus Attack* for camera phones), physical motion (*Dance Dance Revolution* dance pad. Wii Sports) and touch and gestures (*Black and White, Nintendogs*, *Yoshi's Touch and Go*, *Electroplankton, Pac-Pix*) Others use positional sensing/ubiquitous computing and motion sensors (*WarioWare: Twisted*). Games offer rich social systems for interpersonal collaboration and communication (*WOW, EverQuest*). Games even prototype future interface technology such as context-sensitive holograms to encourage cursor exploration (*Star Wars Commando*). The interface for each of these technologies supports a fresh experience.

In addition, games offer an emotional punch that has become a cultural force. The automotive industry already consults racing-game designers on how to make more exciting cars. U.S. politicians such as Howard Dean use web games to teach democracy and increase interest in their campaigns. Games are not only innovating interface design, they are also full of emotional experiments. For example, in *Fable*, the moral values expressed by player decisions change the character into a shining, blonde-haired hero or demonic devil with horns. Although not everyone plays computer games, they command a profound influence on other media and culture. Additionally, the generation raised on games (today's college students all played *Oregon Trail* in grade school) will play more games as adults than their parents did.

Games' distinct emotional processes come from what players like most during play, which informs the functional design of product and software. Unfortunately, the sole emotion goal addressed by usability is to reduce frustration. While important, this falls short of offering strong emotional rewards for accomplishing difficult tasks. Crafting these emotions contributes to user success because, when inspired by rich emotional responses, users will explore and learn more of an application's features, making them more efficient at their jobs. Over-engineering a task by making it too predictable, repetitive, and easy to complete increases the likelihood of boredom and actually reduces satisfaction over the long term. Players and workers experience huge emotional rewards for completing complex challenging tasks. It would be unfortunate if these were eliminated from the work we do as humans.

In HCI, the big mistake used to be fixing clumsy complex features with pretty background graphics. Today, the big mistake lies in believing that user experiences will improve if designers remove frustration points (usability), have the interface do all the work, and reduce the task to a series of trivial steps. Customers already have emotional reactions to their software. Designers must learn to speak this language of emotion from interaction.

The role of a good interface is to prepare users to master something difficult, then allow them to give themselves credit for mastering the difficult skill, while leaving enough ambiguity and challenge to make the task fun. What's next for user-experience design goes beyond refining features to logically support how users perform tasks. Design must also address how features motivate users through affective states to support and refresh the task at hand. Not by making the software unnecessarily complicated, but by doing what games do: support sequential skill-building to achieve complex goals and encourage players to move beyond their points of failure to feel empowered masters of their own destiny. Experience designers are already using Hard Fun, Easy Fun, Serious Fun, and People Fun by embedding emotions from goal-directed, open-ended, purposeful, and social play.

By offering life problems in miniature, games provide important clues to the relationship of emotion in human problem solving, goal achievement, and interaction with other people. Researching games clearly defines the relationships between action and emotion as well as between emotions themselves. In games, many emotions are opposites, happen in sequences, have prerequisites, and share links with others. Some emotions require people, relate to goals, or involve the future or the past. Connecting how gameplay leads to specific emotions establishes a framework for the process of creating more emotional user experiences. The Four Fun Keys Model creates effective analysis and design techniques to identify and create a wider range of emotions. It connects emotions to popular types of gameplay and demonstrates how chains of emotions are embedded or released from different activities. By harnessing play, even productivity software and workflow design can take advantage of the motivating force of game emotions.

Games are the new medium of the 21st century. Unlike any other design discipline, player experiences that unfold over time are more interactive than movies, painting, architecture, industrial design, literature, or fashion. Games are dynamic processes that create dynamic experiences. They are much more than a series of static impressions seen in sequence. Games offer a unique set of emotions from accomplishment and failure. At their core is the ability for players to interact with the content. The experiences that come from this interaction, the ability to create emotions, and the way that this interaction moves will separate games as an art form distinct from movies and other visual arts.

Emotions play an essential role in providing the entertainment value that captivates players. From games, we can learn how to improve the emotions that keep users engaged in activities of different types. Through player interaction and control of events, games promise to become more emotional than movies and other entertainment, but whether these are the same emotions remains to be seen. Games now include more detailed storylines to increase player engagement. However, it is

clear from XEODesign's research that games create strong emotions even without stories, and it is clear that games already create more emotions through interaction with other players in the game world. Many aspects of the storytelling language of cinema apply to games and they will still move players emotionally. However, emotions from the player's goals and the things he or she cares about will likely prove to be stronger. In games, the mechanic is the moral of the story.

Releasing the full emotional potential of games will not be easy. It requires deep understanding of how emotions work, because more is known about crafting emotional entertainment experiences (such as a movie) by offering the viewer empathy with a protagonist engaged in a predetermined sequence of events. Less is known about creating emotion through being the protagonist. Viewing a prerendered video between game levels is less compelling than having the player's actions create an emotionally rich experience itself, but at the moment these mini-movies are easier to design. Going forward, these dramatic tools for creating affect should inspire rather than dictate game design. Eventually the emotional props from these cut scenes will fade from games just as title cards disappeared from old silent pictures once sound technology allowed the actors and the action to speak for themselves.

Psychology, sociology, theater, literature, film, and only recently games have all studied how entertainment engages audiences. Games are unique in that they entertain by creating emotion from interaction. Before XEODesign's research on emotion and the fun of games, none had studied how entertainment creates specific emotions and uses them to focus attention and motivate play. The Four Fun Keys Model creates the four most important sets of emotions for games. Each fun key is a collection of game interactions (game mechanics) that captivate player attention with different series of emotion that make games self-motivating. These emotion cycles create the player experience and separate best-selling games from their imitators. Based on contextual research of people playing their favorite games cross genre, platform, and gender; the Four Fun Keys describe how emotion comes from what is the most fun about games. Entertainment interfaces and products such as Apple's iPod and Microsoft's XBOX 360 already use interaction from each of the Four Fun Keys to unlock powerful emotions to increase user enjoyment and build stronger brand impressions. Interaction that generates a lot of emotion is more memorable and feels like play.

ACKNOWLEDGMENTS

Jane Booth, Brian "Psychochild" Green, Ron Meiners, Steve Meretzky, and Jeff Pobst provided detailed feedback on drafts of this chapter. Hal Barwood, Lura Dolas, Noah Falstein, Bill Fulton, Lee Gilmore, Russ Glaser, Kevin Keeker, Paolo Malabuyo, Marcos Nunes-Ueno, Randy Pagulayan, Patricia Pizer, Kent Quirk, Brian Robbins, Ann Smulka, Mark Terrano, Gordon Walton, and Richard A. Watson all provided useful guidance, suggestions, and examples.

References

4ourty2wo. (2005). Retrieved December 29, 2005, from http://www.4orty2wo.com/casestudy_ai.html

Ahn, L. von, & Dabbish, L. (2004). Labeling images with a computer game. *Proceedings Association for Computing Machinery (ACM) Special Interest Group on Computer-Human Interaction. Conference (CHI).* 319–326. Vienna, Austria. New York: ACM Press.

Bartle, R. 1996 (a). *Hearts, clubs, diamonds, spades: Players who suit MUDs*. MUSE Ltd, Colchester, Essex, UK. Retrieved December 29, 2005, from http://www.brandeis.edu/pubs/jove/HTML/v1/bartle.html

Bartle, R. (2003a). *A self of sense*. Retrieved December 29, 2005, from http://www.mud.co.uk/richard/selfware.htm

Bartle, R. (2003b). *Designing virtual worlds.* New Riders Games. Berkeley, CA: Peach Pit Press.

Boorstin, J. (1990). *Making movies work*. Beverly Hills, CA: Silman-James Press.

Brave, S., & Nass, C. (2002). Editors Jacko, J., & Sears. A. Emotion in human-computer interaction. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications,* (pp. 81–96). Lawrence Erlbaum Associates, Incc., Mahwah, NJ.

Clore, G. C., & Gasper, K. (2000). Feeling is believing: Some affective influences on belief. In N. H. Frijda, A. S. R. Manstead, & S. Bem (Eds.), *Emotions and beliefs: How feelings influence thoughts* (pp. 10–44). Paris/Cambridge: Editions de la Masion des Sciences de l'Homme and Cambridge University Press (jointly published).

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper & Row.

Damasio, A. (1994). *Descartes' error: Emotion, reason, and the human brain*. New York: Quill Penguin Putnam.

Ekman, P. (2003). *Emotions revealed*. New York: Times Books Henry Hold and Company, LLC

Fabricant, R. (2005). *HRV monitor: Creating a guided user experience on handheld devices*. Proc DUX 2005 San Francisco, USA.

Gee, J. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.

Hassenzahl, M., Platz, A., Burmester, M., & Lehner K. (2000). Hedonic and ergonomic quality aspects determine a software's appeal. *Proceedings Association for Computing Machinery (ACM) Special Interest Group on Computer-Human Interation Conference (CHI)*, 201–208, The Hague, the Netherlands.

Hazlett, H. (2003). Measurement of user frustration: A biologic approach. *Proceedings Association for Computing Machinery (ACM) Special Interest Group on Computer-Human Interaction Conference (CHI)*, 734–735, April, 2003

Hjelm, I. (2003). Brain Ball research + design: The making of Brainball. *Interactions, 10*(1) .

Jordan, P. W. (2000). *Designing pleasurable products: An introduction to the new human factors*. London: Taylor & Francis.

Kim, A. J. (2000). *Community building on the Web*. Berkeley, CA: Peach Pit Press.

Lazzaro, N., & Keeker, K. (2004). What's My Method? A game show on games. (pp. 1093–1094). *Proceedings Association for Computing Machinery* (*ACM*) *Special Interest Group on Computer-Human Interaction Conference* (*CHI*), Vienna, Austria.

Lazzaro, N. (2004a, Winter). Why we play games. *User Experience Magazine, 8.*

Lazzaro, N. (2004b). Why we play games: Four keys to more emotion in player experiences. *Proceedings of the Game Developers Conference*, San Jose, California, USA. Retrieved December 28, 2005, from www.xeodesign.com/whyweplaygames.html

Lazzaro, N. (2005a). *Design survey of game designers*. Unpublished manuscript.

Lazzaro, N. (2005b). *Diner dash and the people factor*. Retrieved March 2, 2005, from www.xeodesign.com/whyweplaygames.html

LeBlanc, M., Hunicke, R., & Zubek, R. (2004). *MDA: A formal approach to game design and game research*. Retrieved March 2, 2005, from http://www.cs.northwestern.edu/~hunicke/pubs/MDA.pdf

Malone, T. (1981). Heuristics for designing enjoyable user interfaces: Lessons from computer games. *Proceedings Association for Computing Machinery* (*ACM*) *Special Interest Group on Computer Human Conference* (*CHI*), (pp. 63–68).

Mandryk, R. (2004). Objectively Evaluating Entertainment Technology, *Proceedings Association for Computing Machinery* (*ACM*) *Special Interest Group on Computer-Human Interaction Conference* (*CHI*) (pp. 1057–1058) Doctoral Consortium,Vienna, Austria.

Norman, D. A., (2004) *Emotional design: Why we love* (*or hate*) *everyday things*. New York: Basic Books.

Papert, S. *Hard Fun*. Retrieved December 29, 2005, from http://www.papert.org/articles/HardFun.html

Piaget, J. (1962). *Play, dreams, and imitation in childhood*. New York: Norton.

Sykes, J., & Brown, S. (2003). Affective gaming: Measuring emotion through the gamepad. *Proceedings Association for Computing Machinery* (*ACM*) *Special Interest Group on Computer-Human Interaction Conference* (*CHI*), 732–733.

Tiger, L.(1992). *The pursuit of pleasure* (pp. 52–60). Boston: Little, Brown & Company.

Thorson, E., & Friestad, M. (1985). The effects of emotion on episode memory for television commercials. In P. Cafferata & A. Tybor (Eds.), *Advances in consumer psychology* (pp. 131–136). Lexington, MA: Lexington.

Wright, R., Ritterfeld, L., Mathiak, K. (2006). Does playing violent video games induce aggression? Empirical evidence of a functional magnetic resonance imaging study. In *Media Psychology* Vol 8, No. 1 (pp. 39–60). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Wright, P., McCarthy, J., & Meekison, L. (2003). Making sense of experience. In M. A. Blythe, K. Overbeeke, A. F. Monk, & P. C. Wright (Eds.), *Funology: From usability to enjoyment* (pp. 43–53). Dordrecht, the Netherlands: Kluwer Academic.

For more articles on emotion and game research, see http://www.xeodesign.com/whyweplaygames

All trademarks are the property of their respective holders.

Game References

TBD—Approximately 80 titles w/publisher information to aid students collecting game examples if there is still room in the paper

•35•

MOTOR VEHICLE DRIVER INTERFACES

Paul Green
University of Michigan Transportation Research Institute

Introduction **680**
What Kinds of Telematics Products Are Likely in the Near Term? 680
Chapter Organization 680
What Is the Driving Context in which Users Perform Tasks? **680**
What Kinds of Crashes Are Associated With Telematics? 682
The Scope of the Cell Phone Problem 682
What Kind of Trips Do People Make and Why? 684
Who Are the Users? 685
What Kinds of Vehicles Do People Drive (The Platform Question)? 685
What Measures of Safety and Usability Are of Interest? (The Empirical Measurement Issue) **686**
How Are Telematics Evaluated? **686**
What Design Documents Exist for Telematics? **688**
What Types of Documents Are There? 688
Non-ISO Documents 688
ISO Documents 689
SAE J2364 (The 15-Second Rule) 690
Alliance Principles 691
What Tools and Estimation Procedures Can Aid Telematics Design? **692**
SAE J2365 Calculations 692
IVIS Estimates 692
Closing Thoughts **693**
References **694**

INTRODUCTION

This chapter is written for professionals familiar with human-computer interaction (HCI), but not with the issues and considerations particular to motor vehicles. For non-HCI professionals, reading chapters 1–16 of this text should provide the desired background. For automotive industry interface designers, this chapter should pull together information dispersed throughout the literature.

HCI of interest to motor vehicle designers because of the rapid growth of driver information systems that utilize computers and communications, collectively referred to as "telematics." Included under the umbrella of telematics are navigation systems, cell phones, collision-warning systems of various types, lane departure warning systems, and so forth. These systems can (a) potentially enhance public safety, (b) make transport more efficient (saving time and fuel), (c) make driving more enjoyable, and (d) make drivers more productive.

More specifically, this chapter reviews design documents and evaluation methods for driver interfaces in order to promote safety and ease of use. An underlying theme is that the safety-critical nature of driving leads to significant departures from standard HCI practice, and to some methods and measures that are unique to automotive applications. Since this is a reference handbook, engineering practice receives more attention than scientific theory and, furthermore, given its technology focus, the design of traditional (noncomputer) motor vehicle driver interfaces (such as switches for headlights and windshield wipers) is not covered. For information on those interfaces, see Peacock and Karwowski (1993) or the latest edition of the Society of Automotive Engineers (SAE) Handbook (Society of Automotive Engineers, 2005). Readers interested in additional research literature on telematics should see Michon (1993), Parkes and Franzen (1993), Barfield and Dingus (1997), Noy (1997), or the proceedings from the 2005 Driving Assessment conference (http://ppc.uiowa.edu/driving-assessment/2005/final/index.htm).

What Kinds of Telematics Products Are Likely in the Near Term?

Over the years, several studies have used expert opinions to predict the future of automotive electronics, specifically telematics applications (e.g., Ribbens & Cole, 1989; Underwood, Chen, & Ervin, 1991; Underwood, 1989, 1992; Richardson & Green, 2000). The most extensive information, however, is often contained in proprietary market surveys not for public distribution. The most recent publicly available projections appear in Green and colleagues (2001), summarizing input from 83 senior executives in the automotive industry. Interestingly, those projections coincide with the time period when this handbook is to be published. Table 35.1 lists the mean value of the year estimated when each feature would be introduced into 10% of new luxury vehicles in 2004–2008.

Since these data were collected in August, 2000, and the planning horizon for a new vehicle is three years, estimates through 2004 should reflect real product plans, not speculation about the future. For some features (built-in personal digital assistant [PDA], electronic toll tag), there was not complete agreement on when introduction would occur, as a significant fraction of respondents thought some features would never be introduced. Nonetheless, the table makes the point that overall, these predictions tend to be slightly optimistic, but not unreasonable. Future products are most likely related to driving support systems (various collision warning and avoidance systems, night vision), information systems (cell phones, e-mail/Internet, PDAs), and the use of voice technology. In about a decade or less, the author believes that wearable computers will be a topic of significant interest for driving. (See chapter 15 by Starner, Siewiorek, & Mailagic). In aggregate, these systems could reshape the driver's task from one of real-time control of the vehicle (steering the vehicle and using the throttle to adjust speed) to information management.

Chapter Organization

How might one organize information on telematics? In their classic paper on usability, Gould and Lewis (1985), identified three key principles to be followed when designing products for ease of use:

1. Early focus on users and tasks
2. Empirical measurement
3. Iterative design

These principles not only apply to office applications and web development, but automotive applications as well. In the automotive context, designers need to understand (a) who drives vehicle (users), (b) what in-vehicle tasks they perform, and, most importantly, (c) the driving task, (d) task context, and (e) the consequence of task failures. These topics are the focus of the first part of this chapter.

Secondly, it is important to be able to measure driver and system performance (empirical measurement). That topic constitutes the second part of this chapter.

Surprisingly, there has been little reporting of how iterative design is used in developing driver interfaces, though the approach is used. A great deal of automotive design relies upon following design guidelines, which is the final focus of this chapter.

WHAT IS THE DRIVING CONTEXT IN WHICH USERS PERFORM TASKS?

Scott McNealy, CEO of Sun Microsystems, once said, "A car is nothing more than a Java technology-enabled browser with tires" (Kayl, 2000). He is dead wrong. The author has never heard of anyone claiming, "A computer came out of nowhere, hit me, and vanished." Yet police officers and insurance adjusters hear such claims about motor vehicles every day. Likewise, the author knows of no one who was ever been killed as a consequence of operating a computer at their desk, but the loss of life associated with crashes arising from normal motor vehicle operation is huge.

TABLE 35.1 Estimated Year of Feature Introduction into Luxury Vehicles from Green et al. (2001)
(Bold = estimated to be in 10 percent of 2006 model year luxury vehicles)

Mean Year Estimate	"Never" Responses*	Feature	Description
2004.3	1	**Built-in wireless phone interface**	
2004.4	0	**GPS navigation**	Uses Global Positioning Satellites for location, e.g., Hertz NeverLost
2004.6	1	**Automatic collision notification**	Calls when air bag deploys, e.g., OnStar in GM vehicles
2004.7	2	**Satellite radio**	Nationwide satellite broadcast of 100 channels, e.g., XM, Sirius
2004.8	4	Removable media for entertainment and data	(e.g., iPod, but for entertainment only)
2004.8	0	Email/internet access	
2004.8	10	Built in PDA (e.g., Palm) docking station	PDA is a personal digital assistant
2004.8	0	Adaptive cruise control (ACC)	Scans for traffic and adjust speed to maintain driver set separation
2005.0	1	**Rear parking aid**	Several SUVs (e.g., Murano)
2005.2	1	MP3 support	MP3 is an audio file format
2005.2	0	Bluetooth support	Short-range wireless communication allowing direct link of cellular phones
2005.3	0	**Automatic download of traffic/ congestion information**	Navigation systems in several Honda products
2005.3	2	Blind spot detection and warning	Radar and other systems to detect vehicles to the side and rear
2005.5	1	**Voice operation of some controls**	Jaguar, several Nissan vehicles
2005.6	2	Downloadable software features	
2005.6	2	Downloadable software fixes	
2005.6	3	Forward collision warning	Laser radar systems
2005.7	4	Forward parking aid	Sonar-based short range systems to assess distance ahead
2005.7	6	Lane departure warning	Video systems that scan for lane markings, Infiniti SUVs
2005.7	4	Dual voltage (42/12 volt)	Current power is 12-volt
2005.8	12	Built-in electronic toll and payment tag	Allows travel through toll both without stopping
2005.8	0	General purpose text/data speech capability	
2005.9	7	Large general-purpose display	
2005.9	6	Offboard applications via data link	
2005.9	9	Night vision systems	Infrared-based systems to show objects ahead on a head-up display (HUD), Cadillac
2006.0	16	Black box crash recorder	
2006.0	7	Active suspension	
2006.1	8	Forward collision braking only	Active, not just a warning
2006.2	9	General purpose computer (e.g., AutoPC)	
2006.3	62	Karaoke	Most respondents said never
2006.3	9	Drowsy driver detection	
2006.7	1	42 v electrical system	
2006.8	10	Open electronics bay with utilities	
2006.8	7	Fingerprint or voice-controlled entry	
2006.9	11	Interface to wearable computer	
2007.1	14	Large area HUD	For speedometer, navigation, back up camera, etc.
2007.1	25	Alcohol-impaired driver detection	Most respondents said never
2007.1	1	Brake by wire	
2007.1	4	Drive by wire	
2007.2	6	Hybrid drive-train (electric/combustion)	
2007.6	15	Automatic lane control	Takes over steering
2008.0	20	Forward collision braking and steering	
2008.2	26	All-electric drive-train	

*Respondents had the option of identifying a particular year or selecting never.

According to the World Health Organization (Peden et al., 2004), 0.75 to 1.8 million people die in road traffic crashes each year, or about 3,000 per day. (See also http://www.unece.org/trans/main/welcwp1.html.) Traffic crashes have been ranked ninth in terms of the percentage of disability-adjusted life years lost (Murray & Lopez, 1996b), well ahead of war (16th), violence (19th), and alcohol use (20th). If the current trends continue, by the year 2020, traffic crashes will become the third largest cause of death and disability after clinical depression and heart disease. In 1990, traffic crashes ranked as the ninth biggest killer in the world.

Additional insights come from crash data for the United States, for which reliable, detailed crash statistics are available. According to the U.S. Department of Transportation (2005a), 42,636 deaths occurred in traffic crashes in the United States in 2004. Of them, 19,091 were passenger car occupants; 5,801 were

pickup truck occupants; 4,641 were pedestrians; 4,735 were in SUVs; 4,008 were motorcyclists; 2,036 were in vans; 761 were in larger trucks; 725 were bicyclists; and 808 were in other categories. (For additional information, see the latest edition of Injury Facts [National Safety Council, 2004] and the U.S. Department of Transportation Fatality Analysis Reporting System [FARS] website [http://www-fars.nhtsa.dot.gov/].)

What Kinds of Crashes Are Associated With Telematics?

It is widely accepted that some telematics tasks are distracting and that distraction can lead to crashes. The most commonly cited statistics concerning distraction-related crashes (Table 35.2) were based on data from the Crashworthiness Data System (CDS). CDS is an annual probability sample of approximately 5,000 police-reported crashes involving at least one passenger vehicle that was towed from the scene (out of a population of almost 3.4 million tow-away crashes). Minor crashes (involving property damage only) are not in CDS. CDS crashes are investigated in detail by specially trained teams of professionals who provide much more information than is given in police reports.

Note that distraction-related crashes primarily involved a single vehicle (41%), though rear-end crashes (moving, 10%; stopped, 22%) were also common. Intersection crashes represented another 18% of the total. Crashes tended to peak in the morning rush hour and, to a much lesser extent, in the evening rush. The overwhelming majority of the crashes occurred in good weather (86% clear, 10% rain, 4% snow/hail/sleet), and many occurred at lower speeds (40% at 0–35 mph, 40% at 40–50 mph, 17% at 55–60 mph, and 4% at over 65 mph). Thus, these data suggest that device test scenarios should emphasize situations in which single-vehicle crashes (often run-off-the-road), as well as those involving rear-end collisions into stopped vehicles, are likely.

Other recent evidence appeared in Stutts, Reinfurt, Staplin, and Rodgman (2001) and Eby and Kostyniuk (2004). Table 35.3 shows summary statistics from Stutts and colleagues (2001; CDS files from 1995, the first year for which driver attention was examined, until 1999) concerning the types of distraction involved.

These data, however, should be used with some caution. First, there is a high percentage of unknown and missing data in the set, an outcome typical when distraction is coded. For example, the driver attention status (1 of 360 variables included) was "unknown" for 36% of the cases. Several years have elapsed since the midpoint of this study, and many driver interfaces have changed considerably since the data were collected. For example, according to the Cellular Telecommunications and Internet Association (CTIA; http://files.ctia.org/pdf/CTIA_Survey_Year_End_2006_Graphics.pdf), the number of cell phones in use grew by more than a factor of four between 1997 and 2006. Given that increase in exposure, the percentage of such crashes is likely to be underestimated by that factor using the data from Stutts and colleagues (2001). Unlike external distractions, distractions due to vehicle hardware can be reduced or eliminated by appropriate engineering on the part of manufacturers and suppliers.

The Scope of the Cell Phone Problem

Because telematics devices are new, the data on device-related crashes is limited. This situation has hampered progress in understanding the risks of such devices, especially cell phones, which receive the most attention. At the time this chapter was written, 22 states record whether cell phones are causal factors in crashes, with most just starting to collect this information in the last few years (Sundeen, 2005). The National Conference of State Legislatures data suggested that cell phones were involved in approximately 0.18% of all crashes. Currently, 6% of all drivers are using cell phones at any moment (U.S. Department of Transportation, 2005b); however, given increased feature content (MP3 support, text messaging, broadcast TV), cell-phone use is likely to increase even more than over the last few years.

In terms of fatalities, the data from the National Police Agency (NPA) in Japan is often cited. Admittedly, there is less evidence to assess the quality of that data than there is for U.S.

TABLE 35.2 Distraction/Inattention Crashes by Crash Type from Knipling and Goodman (1996)

Crash Type Row % Column %	Sleepy	Distracted	Looked But Did Not See	Unknown	Attentive	Total
Single Vehicle	5.8	18.1	0.2	31.8	44.1	100.0
	66.2	41.2	0.7	20.6	45.9	30.0
Rear-End/Lead Vehicle Moving	12.7	21.3	3.4	48.3	14.3	100.0
	27.9	9.6	2.0	6.4	2.9	5.9
Rear-End/Lead Vehicle Stopped		23.9	11.4	52.6	11.8	100.0
	—	21.9	13.8	14.1	4.9	12.1
Intersection/Crossing Path		7	17.9	52.8	22.3	100.0
	—	18.1	63.6	39.8	26.6	34.3
Lane Change/Merge		5.6	17.2	41.8	35.3	100.0
	—	1.6	6.7	3.4	4.6	3.8
Head-On	1.0	7.0	8.1	46.4	37.5	100.0
	1.7	2.2	3.5	4.3	5.4	4.2
Other		7.3	9.7	53.5	28.9	100.0
	—	5.4	9.7	11.4	9.7	9.7
Total Crashes	2.6	13.2	9.7	45.7	28.8	100.0
	100.0	100.0	100.0	100.0	100.0	100.0

TABLE 35.3 Crashes for Which Distraction was a Known Factor (8.3 percent of the crashes) (Bold = equipment related distraction)

Distraction	Estimate (%)	Error (%)
Outside person, object, or event	29.4	±4.7
Adjusting radio/cassette/CD	**11.4**	**±7.2**
Other occupant	10.9	±3.3
Moving object in vehicle	4.3	±3.2
Other device/object	2.9	±1.6
Adjusting vehicle/climate controls	**2.8**	**±1.1**
Eating and/or drinking	1.7	±0.6
Using/dialing cell phone	**1.5**	**±0.9**
Smoking-related	0.9	±0.4
Other distractions	25.6	±6.0
Unknown distraction	8.6	±5.3

TABLE 35.4 Mobile Phone and Navigation Systems-Related Crashes in Japan

System	Result	1997	1998	1999
Cell Phones	Injuries	2,095	2,397	2,418
	Deaths	20	28	24
Navigation Systems	Injuries	117	131	205
	Deaths	1	2	2

TABLE 35.5 Cell Phone and Navigation Crashes per Year in Japan

Crashes	1997	1998	1999	2000	2001	2002	2003	2004
Cell phones	2297	2648	2583	1453	3040	2847	2597	1868
Navigation systems	127	146	219	190	1415	1307	1098	1253

federal or state data. Further, it is suspected that even when investigated, mobile phone use is underreported, because reporting involves drivers admitting to something inappropriate. Tables 35.4 and 35.5 show the annual crash statistics for phones and, for perspective, navigation systems. The decreases in the number of cell-phone crashes after 1999 and 2003 reflect new legal restrictions on cell phone use.

Using these data, cell-phone-related deaths in the United States for the year in which this chapter was written (2005) can be estimated. (See Wierwille, 1995.) The average number of cell-phone-related deaths in Japan was about 25 per year for 1997–1999. In determining this estimate, one must realize that a random variation exists in deaths per year, and the number increased over that period. On average, the number of traffic deaths in the United States is about four and a half times that of Japan. Assuming everything else is equal, this would suggest at least 112 (=25 * 4.5) cell-phone-related deaths per year in the United States for 1997–1999. If the number of calls per driver and the mean call duration are assumed unchanged (they have actually increased by 30%), then the change in exposure of the population between 1998 and 2005 is simply due to the increase in the number of phones in existence, roughly 3.19 times. Using the CTIA data to determine the number of users and 112 deaths in 1998 as the base estimate, the total for 2005 cell-phone-related crashes in the United States is 357 deaths (=112 × 3.19); this number should continue to increase at 15% or so per year, the current growth rate of cell phone subscribers. Other sources suggest the number of cell-phone-related deaths is much higher (Pena, 2007).

By way of comparison, about 120 children and small-stature women were killed in front-passenger air-bag crashes, and about 200 people in Firestone/Explorer rollover crashes over several years. Both of these situations have led to major changes in federal requirements and product engineering changes, as well as significant financial losses to several manufacturers (Green, 2001c). In response to the Firestone situation, the U.S. Senate passed the TREAD Act (http://www.driveusa.net/treadact.htm) three weeks after it was introduced. Hence, designing telematics interfaces to assure safety and usability is a topic worthy of significant effort.

Supporting the crash statistics, the most-often-cited study of cellular-phone use is Redelmeier and Tibshirani (1997; see also Redelmeier & Tibshirani, 2001.) They examined data for almost 700 drivers who were mobile-phone users and were involved in motor-vehicle crashes that resulted in substantial property damage. Each driver's mobile-phone records for the day of the crash and the previous week were examined. Redelmeier and Tibshirani reported the risk of a crash was 4.3 times greater when a mobile phone was used than when it was not. Interestingly, hands-free units had a greater risk ratio (though not significant) than hand-held units (5.9:1 versus 3.9:1). Other data (Koushki, Ali, & Al-Saleh, 1999; McEvoy et al., 2005; Violanti & Marshall, 1996) suggested similar risk ratios. Other human factors data show drivers taking longer to respond to brake lights of lead vehicles, departing from the lane more often, and exhibiting other undesired characteristics (Horrey & Wickens, 2004) while using a cell phone.

If phone use is a concern, which tasks lead to crashes? Table 35.6 shows task-specific crash statistics for cell phones and, for comparison, navigation systems. These NPA data identify four primary tasks: (a) receiving a call, (b) dialing, (c) talking, and (d) other. What other represents is unknown. It is also unknown if talking includes voice services (voice mail, bank by phone, etc.). Notice that NPA reported the number one cause of crashes was receiving a call. Most people, either at work or at home (and while driving, even in hazardous traffic situations), immediately (within 1–4 seconds) answer the phone when it rings, almost regardless of the situation (Nowakowski, Friedman, & Green, 2001). (An exception is at dinnertime in the United States, when telemarketing calls are common.) For example, one could be in his or her office talking to a very important person such as the

TABLE 36.6 Driver Tasks and Crashes (January–November, 1999) from NPA Data

Cell Phone		Navigation System	
Task	Crashes	Task	Crashes
Receiving Call	1077		
Dialing	504	Looking	151
Talking	350	Operating	46
Other	487	Other	8
Total	2418		205

President of the United States. If the phone rang, many people might ask the President to wait a moment while they answered the incoming call, though it is highly likely someone more important might be calling. In fact, many people now have call waiting so they can accept additional incoming calls while on the phone. Logically, answering the phone should not usually be more important than driving, but people behave otherwise out of habit.

Not only is the answering task a problem because of its immediacy, but also sometimes because of its visual demands. Many drivers use portable phones that might be in a jacket pocket on the seat or in a purse. Thus, in answering the phone, the driver's first task might be to search for it. Although hands-free systems can reduce the duration of the disruption, they do not eliminate the disruption itself or its incorrect prioritization.

In the Japanese data, the second most common cause of crashes was dialing. Clearly, going to a hands-free interface can reduce the visual and manual demands, though not the cognitive demands of dialing; however, the current evidence is that there is no difference in crash risk when hands-free phones are provided. It could be that providing a hands-free phone makes it easier to dial, so hands-free phone users engage in more conversations while driving. Another possibility is that seeing a phone being held is a warning to other drivers of a potentially distracted driver.

Finally, a substantial number of calls were associated with just talking on the phone. While some distractions associated with talking on the phone are manual (such as holding the phone), most distractions are cognitive, with drivers focusing on the conversation, rather than on driving. It is this aspect of cell phone use that is commonly observed by drivers. Current wisdom in the U.S. is that if a person is weaving between lanes and it is night, they are drunk. If they are weaving and it is during the day, they are on the phone.

Talking on the phone is different from talking to a passenger, especially an adult in the front seat. Often, that adult behaves as a co-driver, looking both ways at intersections and checking the mirrors during lane changes, and speaking less in higher risk situations (Drews, Pasupathi, & Strayer, 2004; Crundall, Bains, Chapman, & Underwood, 2005). When drivers move their heads to scan an intersection or expressway entrance or exit, passengers often stop talking. People on the phone have no knowledge of the driving situation, and just keep talking. Technology could assist in reducing the scope of this problem, for example by providing cues to inform callers of the driving situation and drivers of the call duration or their driving performance.

Recognizing the risks of cell phone use, many countries (Australia, England, Israel, Italy, Switzerland, and Spain) permit only hands-free operation. As was noted earlier, however, hands-free use is essentially no different in risk from handheld operation. Many states are also considering restrictions on cell phone use while driving, with the exception of making 911 and other emergency calls. Obtaining passage of such laws has proven difficult.

The author believes that a complementary approach is to implement an in-vehicle workload manager (Michon, 1993; Green, 2000b, 2004; Hoedemaeker, de Ridder, & Janssen, 2002; Piechulla, Mayser, Gehrke, & König, 2003). As initially conceived, such systems would use data from four sources: (a) the navigation system (such as lane width and radius of curvature), to assess the demands due to road geometry, (b) the adaptive cruise control system (headway and range rate to vehicles ahead), to assess traffic demands, (c) the traction control system, to assess road surface friction, and (d) the wipers, lights, and clock, to assess visibility. This information—along with information on the driver (e.g., age) and the specific visual, auditory, cognitive, and motor demands of each in-vehicle task—could be used to schedule the occurrence of in-vehicle tasks. Thus, when driving on a curving road in heavy traffic in a downpour, incoming mobile phone calls could be directed to an answering machine and 30,000-mile maintenance reminders could be postponed. When the driving task demand is low, drivers could have access to a wide range of functions. Being able to reliably predict the momentary workload of driving, however, has proven to be very difficult.

As an alternative, it is useful to think more broadly about when the driver is likely to be overloaded and, as some vehicle engineers have realized, that often occurs when the driver is maneuvering or about to maneuver. Maneuvering includes (a) changing lanes, (b) merging onto an expressway, (c) turning at an intersection, (d) parking, (e) braking in response to a lead vehicle, (f) accelerating from a traffic light, and so forth. These situations are much easier to identify than the overload situations described earlier and, furthermore, are situations when drivers are less likely to desire additional tasks (such as responding to an incoming call) and would likely lead to much greater market acceptance of a workload manager (Eoh, Green, Schweitzer, & Hegedus 2006).

What Kind of Trips Do People Make and Why?

Every five years, the U.S. Department of Transportation conducts the National Household Travel Survey to obtain travel data for the United States (Hu & Reuscher, 2004). In the most recent data from 2001, an average person in the United States traveled about 14,500 miles per year, making four trips per day. On average, people drove about 40 miles per day, with most of the miles (about 35) covered in a personal vehicle. Keep in mind that these are averages, and that public transit (including school buses) prevalent in urban areas accounts for only 2% of all trips. The travel situation is likely to be different for more urbanized countries (Japan, most of Europe) as well where public transit is more prevalent.

According to the 2001 data (Table 35.7), the most common reason for travel is family and personal business, which includes shopping, running errands, and dropping off and picking up others, accounting for almost half of the trips.

These and other data (on trip distances, travel speeds, time of day, etc.) in the National Household Travel Survey provide

TABLE 35.7 Summary of Trip Purposes

Purpose	% Person Trips
Family & personal business	44.6
Work	14.8
Social & recreational	27.1
School & church	9.8
Work-related	2.9
Other	0.8

background information on the context for which telematics devices should be designed and may suggest scenarios that should be used in evaluating safety and usability.

In contrast to the emerging understanding of the driving task, less is known about the real use of in-vehicle devices while driving, in particular the frequency and duration of use.

Who Are the Users?

Almost any adult has the potential of driving, with the determination being made by completing simple driver licensing requirements. Thus, in some ways, the driving population represents the population of candidate users for office computer systems. For most states, the percentage of drivers of either gender is within 1% of being equal (Highway Statistics 2003, 2004a). Data with regard to age are shown in Fig. 35.1. Notice that the percentage of the population that is licensed hits 80% at age 21 and remains at that level until the mid-70s. Thus, in designing in-vehicle systems for motor vehicles, few adults can be excluded, which differs from the design of office computer systems, where the emphasis is on the working population (generally less than 65 years old). Further, because of a wide age range and other differences, significant differences in individual performance can be expected. For example, in UMTRI telematics studies, older drivers typically required one and a half to two times longer to complete tasks than younger drivers (Green, 2001d). This fact, along with the requirement to design and test for the reasonable worst-case drivers, makes testing drivers over age 65 imperative.

In contrast to users of computers, operators of motor vehicles must be licensed. In the United States, the process of becoming a licensed driver begins with obtaining a copy of the state driving manual and learning the state's traffic laws. Candidates must also pass vision tests (see http://www.lowvisioncare.com/visionlaws.htm) and take a test of rules of the road to obtain a learner's permit, often on their 16th birthdays. Consistent with the increasingly common practice of graduated driver licensing, the learner can drive at restricted times with adult supervision, must generally complete a driver's education class (which often includes gory crash movies to convince teenagers not to drink and drive and to wear seat belts), and after a few years and an on-the-road test, is licensed to drive. (For details, see http://www.highwaysafety.org/laws/state_laws/grad_license.html, http://www.insure.com/auto/teenstates.html, and http://www.hsrc.unc.edu/pubinfo/grad_overview.htm.) The rationale for graduated licensing is to provide new drivers with more experience under less risky conditions. To put this in perspective, Mayhew, Simpson, and Pak (2003) show that crash rates per 10,000 novice drivers drop dramatically with time, being about 120, 100, and 70 after one, three, and six months of being licensed. Similarly, recognizing the increased risk of elderly drivers, some states have special renewal procedures for them (IIHS, 2006).

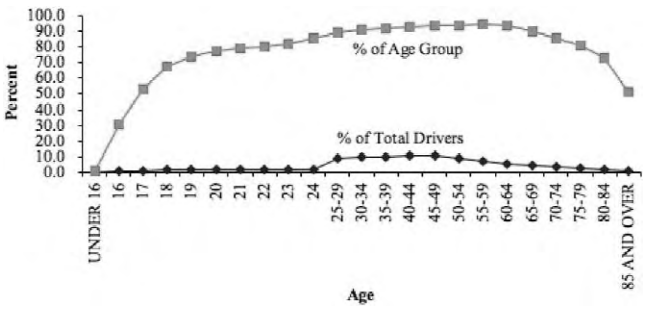


FIGURE 35.1. Distribution of driver age groups developed from U.S. Department of Transportation data.

In the United States, obtaining a commercial driver's license, which is needed to drive buses, large trucks, and other vehicles, is a more complex process. Most candidates either obtain (a) on-the-job training, (b) training integrated into their lifestyle (using machinery on a farm), or (c) training at truck driving schools (Sloss & Green, 2000). That population tends to be older than the working population as a whole, and is predominantly male.

Driver licensing practice varies from country to country. In Japan, for example, the failure rate for the basic licensing exam is much higher than in the United States, and there is much greater use of special schools to train drivers.

What Kinds of Vehicles Do People Drive (The Platform Question)?

For computers, people are concerned about the (a) brand, (b) amount of memory, (c) processor speed, (d) capacity of the hard drive, (e) type and version of operating system (Windows, Mac, or Linux), (f) type and version of browser, and so forth. The hardware rarely moves very far from its initial location; however, the content of individual computers is in a state of flux, being constantly updated over a life span of just a few years. Fortunately, the physical interface is fairly consistent—a QWERTY keyboard, mouse, and often a 17-inch or 19-inch monitor. The on–screen "desktop" is a more flexible space than the motor vehicle instrument panel.

Unlike personal computers, a motor vehicle is almost completely identified by its make, model, and year. Updates over an average 11-year life span are rare. In most countries (at least where there is left-hand drive), many aspects of driving are fairly consistent: the (a) input devices (steering wheel, brake, and throttle), (b) method of operation, (c) location, and (d) primary displays (windshield and mirrors). In contrast, there is no consistency in the controls or displays for telematics interfaces. Furthermore, although new motor vehicle models are offered once per year, major changes typically occur every five years. Computer software and hardware model upgrades occur almost continually, with a three-year-old computer generally being considered obsolete. Thus, the hardware life cycles of the two contexts are quite different.

According to the U.S. Department of Transportation, 237 million motor vehicles were registered in the United States in 2003 (Highway Statistics 2003, 2004b). Of them, 57.3% were automobiles, 40.1% were trucks, 2.3% were motorcycles, and 0.3% were buses. In less-developed countries, motorcycles make up a much larger fraction of the vehicle fleet.

For 2004, North American car and truck production was 16.9 million vehicles (Tierney, 2005). Many people do not realize that the United States now produces more trucks than cars. The pre-

TABLE 35.8 Top 20 Vehicles in U.S. (Sales in 2004) from Wards Auto World
(Bold = trucks)

Rank	Make	Model	Sales
1	**Ford**	**F-Series**	**901,931**
2	**Chevrolet**	**Silverado**	**680,768**
3	Toyota	Camry/Camry Solara	426,990
4	**Dodge**	**Ram Pickup**	**426,289**
5	Honda	Accord	386,770
6	Honda	Civic	309,196
7	Chevrolet	Impala	290,259
8	**Ford**	**Explorer**	**276,303**
9	Chevrolet	Malibu	268,017
10	Ford	Taurus	248,148
11	Toyota	Corolla	243,208
12	**Dodge**	**Caravan**	**242,307**
13	Nissan	Altima	235,889
14	**GMC**	**Sierra**	**213,756**
15	Ford	Focus	208,339
16	Chevrolet	Cavalier	195,275
17	**Chevrolet**	**Trail Blazer**	**193,862**
18	**Chevrolet**	**Tahoe**	**186,161**
19	**Ford**	**Escape**	**183,430**
20	**Jeep**	**Grand Cherokee**	**182,313**

dominance of trucks is due the popularity of minivans, sport utility vehicles (SUVs), and light trucks for personal use, not the production of heavy trucks for commercial use. Table 35.8 summarizes U.S. sales, including vehicles made in Japan, which comprise a significant fraction of U.S. sales. Trucks predominate sales. Thus, as designers think about packaging telematics applications, trucks are a likely platform.

WHAT MEASURES OF SAFETY AND USABILITY ARE OF INTEREST? (THE EMPIRICAL MEASUREMENT ISSUE)

As was noted earlier, usability is difficult to achieve without empirical measurement. First impressions suggest that the measurement of usability of office computer and web applications, and the measurement of the usability of telematics applications, are quite similar. In an office, one measures task completion time, errors, and ratings of ease of use. A typical usability lab has (a) a one-way mirror, (b) cameras, (c) video editing equipment, (d) audio mixers, and (e) at least two rooms, one for the subject and one for an experimenter. (See chapter 57.)

In a typical laboratory for examining telematics, the same measures may be obtained; however, other driving-specific measures, as listed in Table 35.9, may also be obtained, especially in driving simulators and on-the-road tests. (See Green, 1995a, b, c.) In addition, a host of other performance measures often used in non-HCI contexts are also often collected: (a) ratings of workload (Cooper-Harper ratings, NASA TLX), (b) measures of situation awareness, and (c) measures of object and event detection (pertaining to traffic). (See Roskarn et al., 2002; Tijerina, Angell, Austria, Tan, & Kochhar, 2003; Johansson et al., 2004.) Because the user is engaged in concurrent performance of a manual control and information management tasks, the situation is in some ways is similar to that described by Landry in the aerospace chapter. (See chapter 36.)

TABLE 35.9. Some Driving-Specific Usability Measures

Category	Measure
Lateral	Number of lane departures
	Mean and standard deviation of lane position
	Standard deviation of steering wheel angle
	Number of larger steering wheel reversals
	Time to line crossing
	Steering entropy
Longitudinal	Number of collisions
	Time to collision
	Headway (time or distance to lead vehicle)
	Mean and standard deviation of speed
	Speed drop during a task
	Heading entropy
	Number of braking events over some g threshold
Visual	Number of glances
	Mean glance duration
	Maximum glance duration
	Total eyes-off-the-road time

A major challenge in assessing the safety and usability of telematics is dealing with the trade-offs that drivers naturally make. The impression is that when preoccupied with an in-vehicle task, drivers lose awareness of the driving context—that is, situation awareness. Drivers attempt to compensate by slowing down (to make driving easier), allowing for larger headways and, if very preoccupied, paying less attention to steering (so lane variance and the number of lane departures increase); however, drivers can respond in strange ways. For example, if asked to use two different in-vehicle systems, one of which is not well designed, they might attempt to maintain equal performance on both: slow down more for the more difficult interface, but compensate by having better steering performance for the poorer interface. Assessment is difficult because the trade-off functions for all of these measures are unknown. One strategy used to overcome the trade-off problem is to minimize the opportunity for trade-offs. For example, this might include using cruise control to fix the speed (and in some cases, headway) and provide incentives and feedback to maintain driving performance, so only task time and errors might trade off.

HOW ARE TELEMATICS EVALUATED?

The assessment of mobile applications often occurs in contexts other than simulators and real roads as suggested above. Table 35.10 provides a summary of the contexts and their strengths and weaknesses. (See Green, 1995a, for additional details.)

Over the last decade, enormous improvements have been made in the quality of the tools available for human factors evaluations of in-vehicle evaluations. These include:

1. Reductions in the size of video cameras (to that of a postage stamp) and their cost.

TABLE 35.10. Evaluation Contexts

	Method and Equipment	Advice and Comments
Focus groups	• Groups of eight to 12 people demographically similar to customers sit around a table and discuss a product or service guided by a facilitator • Camera is often behind one-way mirror • Generally done in multiple cities (one or two groups/city) • Often conducted by a marketing firm	• Useful in getting ideas for product concepts, but not predictions of the safety or usability of new products because the products are never used • Approach might be used by manufacturers when a usability test might be more appropriate • Generally no quantitative data • Essential to report actual quotes from participants, not what the facilitator recalls
Clinics	• Customers in various cities are given the opportunity to experience a new product and its competition, often two or three vehicles, side by side • Customers say which product or feature they prefer	• Only exposes users to a limited number of options • Performance data often not collected • Approach is commonly used by industry • Because the results are highly proprietary, published studies are rare
Part task simulation	• A sample of users operate the device (e.g., computer-simulation of a new radio) and user task times, errors, and comments are recorded • Test facility is not sophisticated	• Not done that often • Relatively less common now than in the past as simulators have improved and instrumented vehicles have become more common
Driving simulator	• Typically driving simulators are fixed base (no motion) and cost $25,000 to $250,000 each • simulators at manufacturers tend to be in 1–3 million dollar range, though some are much more (e.g., Ford is about $10,000,000). • 1–5 projectors with total 40–210 degree forward field of view, real vehicle cab, steering system with torque feedback, and realistic sound • rear image may be projected or mirrors may be replaced with small LCDs • For advice, see Green, Nowakowski, Mayer, and Tsimhoni, 2003; Green, 2005	• Operation requires considerable experience • Simulator sickness is a major problem • Each experiment requires construction of a test road/world and scripting the behavior of vehicles and pedestrians • Facility can require considerable space (e.g., 1000 square feet) • Generally requires large number of fixed small (lipstick or smaller) cameras • Best-known vendors in U.S are Systems Technology Inc. ($5,000–$25,000, http://www.systemstech.com/content/view/23/39/), GlobalSim Corporation, ($75,000–$150,000, http://www.drivesafety.com/), and OKTAL in Europe (http://www.scaner2.com/php/)
Instrumented vehicle on test track or public roads	• Production vehicle (usually a car and often in the past, a station wagon) is fitted with cameras aimed at driver, forward scene, instrument panel, and lane markings, and with sensors for steering wheel angle, brake pressure, speed, and headway • Eye fixation system may also be provided • System of interest is also installed • In a typical experiment, the driver is asked to follow a test route while a back seat experimenter operates the test equipment	• Typical cost is at least $100,000–$250,000 though some low-cost systems may utilize a single camcorder on a vertically-mounted curtain rod in the back seat aimed at the instrument panel • Problem in the past has been finding enough space and power for the equipment, which has been solved by laptops and lipstick cameras • Commonly, instrumented research test vehicles are used, though there is increasing use of systems that can be installed in the subject's own vehicle
Operational field test	• Compact instrumentation is installed in a fleet of vehicles (10–50) • Each vehicle is borrowed by a potential user for a week, a month, or even a year • Driving performance is surreptitiously recorded by the vehicle • Unlike an instrumented car, continuous video is not recorded • In addition to data recorded by the instrumented vehicle, GPS-determined location is also recorded • Vehicles are periodically polled for data (and data is automatically dumped) by an independent digital cellular phone • Test is confined to a single metropolitan area	• Each test requires unique instrumentation • Tests are very expensive ($10,000,000 to $40,000,000) and can only be conducted with significant government support • Experiment generally lasts several years • Planning stage for experiments takes several years • At any given time, there may only one operational field test in progress in the U.S.

2. Significant improvements in the fidelity, along with significant reductions in the cost, of wide field-of-view driving simulators.
3. Electronic innovations that allow packaging compact instrumentation systems in vehicles and that reduce power consumption (and heat generation).
4. GPS systems for precise tracking of vehicle location.
5. Digital cellular phones for remote downloading of vehicle data.

Development of better, lower-cost eye-fixation systems is still needed, though the leading vendors of hardware for driving research. Seeing Machines, the manufacturers of FaceLab (http://seeingmachines.com/), and Smart Eye (http://www.smarteye.se) are making progress. Challenges include their accuracy and ability to hold calibration, dealing with glasses (common for older drivers), and operation in bright sunlight.

WHAT DESIGN DOCUMENTS EXIST FOR TELEMATICS?

What Types of Documents Are There?

To establish that a product is safe and easy to use, empirical measurement is important. Beyond verification, findings from such measurements can be fed back to the design team to improve the design.

Design is also guided by product and service specifications. In the case of office applications, style guidelines exist for various platforms and applications. (See chapter 47 by Mayhew for related information on requirements specification.) For telematics applications, publicly available, product-specific interface guidelines do not exist but there are other types of important written materials (Green, 2001a, 2001b; Schindhelm et al., 2004). Automotive design documents fall into five general classes: (a) principles, (b) information reports, (c) guidelines, (d) recommended practices, and (e) standards. Principles give high-level recommendations for design and are similar to those found in office HCI applications, such as "design interfaces to minimize learning."

Information report is a term used by SAE to refer to a compilation of engineering reference data or educational material useful to the technical community. Information reports do not specify how something should be designed, but provide useful background information.

Guidelines give much more specific advice about how to design an interface element. For example, guideline 9 in chapter 7 of Green (1995) stated, "Turn displays should show two turns in a row when the turns are in close proximity" (p. 41), where close proximity means 0.1 miles apart or less. The impact of guidelines can depend on the issuing organization. For example, automotive design guidelines written by research organizations have no real authority. Guidelines written by the International Organization of Standardization (ISO), while technically voluntary, can become requirements, because in some countries, type approval (approval for sale) requires compliance with ISO guidelines. For vehicle models sold worldwide, global manufacturers find building common vehicle systems that comply with ISO standards to be less costly than building noncompliant, country-specific systems. In Japan, the Japan Automobile Manufacturers Association (JAMA, 2004) has a set of guidelines for navigation systems. Although theoretically voluntary, "requests" from the National Police Agency make the JAMA guidelines a requirement for all original equipment manufacturers (OEMs).

Recommended practice, a term used by SAE, refers to methods, procedures, and technology that are intended as guides to standard engineering practice. The content may be of general nature, or may present data that have not yet gained broad engineering acceptance. Common practice is to follow a recommended practice, if possible. It is recognized that a product liability action against a product (especially in the United States) is extraordinarily difficult to defend if the product design deviates from recommended practice.

A standard specifies how something must be done and includes broadly accepted engineering practices, or specifications for a material, product, process, procedure, or test method. In the case of SAE, a standard is technically voluntary because SAE has no enforcement powers. Even more so than with a recommended practice, a product not complying with an SAE standard is almost not defendable in a product liability action, and is unlikely to be purchased from a supplier by an OEM.

Non-ISO Documents

Table 35.11 provides a summary of the design document activities to date excluding those of the ISO (described later). As indicated in the table, the EU guidelines are quiet brief and are merely statements of very general principles (for example that interfaces should be simple to operate). Others contain a limited set of specifics (Alliance, JAMA, SAE J 2364, and J2365, TRL), and others contain significant design details (Battelle, HARDIE, UMTRI). Readers should not take the view that the newer or longer sets of guidelines are necessarily better. Representatives from industry (with academic involvement for SAE efforts) developed the Alliance, EU, JAMA, and SAE guidelines. Contractors (Battelle and UMTRI) working for the U.S. Department of Transportation developed the guidelines bearing their names. The TRL and HARDIE guidelines were contracted efforts in Europe.

See http://www.umich.edu/~driving/guidelines/guidelines.html. for unofficial electronic copies. (To avoid copyright problems, only draft and not final versions have been posted for SAE and ISO documents.)

All of these guidelines are voluntary except for the JAMA guidelines, as noted earlier. In the United States, the most important guidelines from the perspective of regulation are SAE Recommended Practices J2364 and J2365, especially J2364, and the Alliance guidelines. They are all discussed later in this chapter. Many manufacturers and suppliers are complying with them.

The guidelines developed by the Alliance, EU, and JAMA are being revised and that process is expected to continue for several years. For updates, readers are therefore advised to see either the authoring organization or the previously noted UMTRI website.

TABLE 35.11. Major Non-ISO Telematics Guidelines and Recommended Practices

Common Document Name	Reference	Size (pages)	Comments
Alliance guidelines	Alliance of Automobile Manufacturers (AAM) (2003 June 17); version 3	67	Restatement of EU principles, plus considerable details and rationale, to be used by almost all manufacturers in U.S.; key sections are principles 2.1 and 2.2, which still need development; still being updated by AAM
Battelle guidelines	Campbell, Carney, and Kantowitz, (1997)	261	Voluminous document with references to interface design, heavy on trucks. User interface has been said to have a Windows OS flavor, includes physical ergonomics information (e.g., legibility, control sizes) which are not included in the UMTRI guidelines.
EU guidelines	Commission of the European Communities (1999)	2	Mostly "motherhood" statements. Some revisions are expected.
HARDIE guidelines	Ross, Midtland, Fuchs, Pauzie, Engert, Duncan, Vaughan, Vernet, Peters, Burnett, and May (1996)	480	Early set of European guidelines, less data than UMTRI or Battelle.
JAMA guidelines	Japan Automobile Manufacturers Association (2004)	15	First set of detailed design guidelines for driver interfaces. These guidelines are voluntary in Japan but followed by all OEMs there and sometimes by aftermarket suppliers. Some aspects are particular to Japan. Device location restrictions are important.
SAE J2364 ("15-second rule")	Society of Automotive Engineers (2004, August)	13	Specifies the maximum allowable task time and test procedures for navigation system tasks performed while driving for systems with visual displays and manual controls; also describes an interrupted vision (visual occlusion) method as well; See also SAE J2678.
SAE J2365 (SAE calculations)	Society of Automotive Engineers (2002, May)	23	Method to compute total task time for tasks not involving voice used early in design to estimate compliance.
TRL checklist	Quimby (1999)	18	Simple check list
UMTRI guidelines	Green, Levison, Paelke, and Serafin (1993)	111	First set of comprehensive design guidelines for the U.S. Includes principles, general guidelines, and specific design criteria with an emphasis on navigation interfaces.

ISO Documents

Much of the ISO activity over the last decade has occurred under the auspices of International Organization of Standardization Technical Committee 22, Subcommittee 13 (ISO TC 22/SC 13-Ergonomics Applicable to Road Vehicles, in particular, Working Group 8 [WG8—Transport, Information, and Control Systems or TICS]; Green, 2000a). WG8 has about 50 delegates from the major vehicle producing nations, with the most 15 active members appearing at meetings held 2 to 3 times per year, usually in Europe.

Table 35.12 shows the standard and technical reports developed (or in progress) by Working Group 8 that pertain to telematics. (For the complete list, see ISO, 2007.) Most of the standards can be quite general, sometimes not containing the detail found in the Battelle, HARDIE, or UMTRI guidelines. For a variety of reasons, they tend to emphasize measurement methods and organization over specifications and safety limits. To promote international harmonization, national standards organizations, technical societies (e.g., SAE), and government organizations (e.g., U.S. Department of Transportation) often permit ISO standards to supersede their own standards, so ISO standards are very important.

Note: ISO documents follow a very-well-defined, three-year process through several stages (Preliminary Work Item [PWI], Committee Draft [CD], Draft International Standard [DIS], Final Draft International Standard FDIS], and International Standard [IS]) as they are passed from the working group to the subcommittee to the technical committee, and finally, to the secretariat for review and approval. The major hurdles are the working group and subcommittee, where passage requires two-thirds of the nations participating. The emphasis of this process is on building a voluntary consensus (ISO, 2003). Some items that are more informational in nature become technical reports instead of standards. Because of the limited number of experts available, WG8 is very selective in adding items to its work program.

TABLE 35.12. ISO TC 22/SC 13/WG 8 Work Program

Effort	Summary	Status
Dialogue management principles and compliance procedures	Provides high-level ergonomic principles (compatibility with driving, consistency, simplicity, error tolerance, etc.) to be applied in the design of dialogues that take place between the driver of a road vehicle and an in-vehicle information system while the vehicle is in motion. Provides general directions on how to test for compliance.	Std 15005
Specifications and compliance procedures for in-vehicle auditory presentation	Provides requirements for auditory messages including signal levels, appropriateness, coding, etc., along with compliance test procedures	Std 15006
Measurement of driver visual behavior	Generally describes video-based equipment (cameras, recording procedures, etc.) and procedures (subject descriptions, experiment design parameters, tasks, performance measures, etc.) used to measure driver visual behavior	Std 15007, part 1 and TS part 2
Legibility (visual presentation of information)	Provides requirements for character size, contrast, luminance, etc. and specifies how they are to be measured	Std 15008
Warning system messages	This state-of-the-art literature review (circa 2002) of warning systems covers topics such alarm theories; the design of visual, auditory, and tactile warnings; redundancy; etc. The report contains summaries of a significant number of studies.	TR 16352
Message priority	Provides two methods for determining a priority index for in-vehicle messages (e.g., navigation turn instruction, collision warning, low oil) presented to drivers while driving. For one method, priority is based on criticality (likelihood of injury if the event occurs and urgency (required response time)), determined on four-point scales by experts.	TS 16951
Suitability of TICS while driving	Generally describes a process for assessing whether a specific TICS, or a combination of TICS with other in-vehicle systems, is suitable for use while driving. It addresses: a user-oriented TICS description and context of use, TICS task description and analysis, assessment, and documentation.	Std 17287
Occlusion method to assess distraction	Describes how the visual demand of a display can be assessed by periodically blocking (occluding) the driver's view of the display. Includes requirements for number and training of subjects, test hardware, viewing and occluded periods, and two metrics for data analysis.	Std 16673 (in press)
Lane change test	Proposes a procedure for testing the demand of telematics devices using a PC-based driving simulator. Subjects perform a number of lane changes, some of which occur while using an in-vehicle device. The test is based on results of the Advanced Driver Attention Metrics (ADAM) project. Complements SAE and AAM procedures.	NWI 26022

SAE J2364 (The 15-Second Rule)

Of the design documents in the literature, few have generated as much discussion as SAE Recommended Practice J2364, also known as "The 15-Second Total Task Time Rule" or "15-Second Rule." SAE J2364 establishes two procedures for determining if a navigation-system-related data entry task involving visual displays and manual controls is excessive while driving (Society of Automotive Engineers, 2004a, b; Green, 1999c). The practice applies to OEM and aftermarket products. The practice does not and should not apply to voice interfaces or passenger operation because the task demands are fundamentally different. There is no reason why the requirements do not make sense for similar tasks for other systems given its performance basis.

In developing this practice, criteria were selected to be related to crash risk, likely to lead to design improvements, and easy to measure. Some suggested this document should contain design criteria, such as a specification for a maximum number of items on a menu. It became clear, however, that such a design practice needed a performance basis, and specifying a design standard would constrain innovation, so a performance-based practice was developed instead. Many measures were then proposed, with eyes-off-the-road time being most popular. The logic behind this measure is simple. If drivers are not looking at the road while driving, they are more likely to crash, with the likelihood of a crash increasing with increased eyes-off-the-road time. As shown in Green (1999d), this is reflected in the following equation:

$$\#\ \text{U.S. deaths in 1989} = *\text{market.penetration.fraction}* [-.133 + (.0447 * (\text{mean glance time})^{1.5} (\#\ \text{of glances}) (\text{frequency}))]$$

where

market.penetration.fraction is the fraction of vehicles with a system (10% ≥ .1) *mean glance time* is the number of seconds drivers looked away from the road *# of glances* is the number of times the device is looked for each use sequence *frequency* is the number of use sequences per week.

As an example, a task—say, entering an address—could have a mean glance duration of 2.7 seconds, require an average of 27.5 glances per entry, and be performed twice per week.

Unfortunately, eyes-off-the-road time is time-consuming, expensive to measure, and requires specialized equipment and skilled personnel. It also requires a fully functional system installed either in a simulator or test vehicle, something that is only available late in design. Invariably, problems are identified so close to production that few, if any, changes can be made. One of the key lessons from the Gould and Lewis design principles from HCI is that feedback from early on in the design process is critical.

A review of the literature (Green, 1999d) found that task time while driving was highly correlated with eyes-off-the-road time. Looking at in-vehicle systems is time shared with driving, so the more the driver needs to look, the longer the task will take. Further, dynamic (on-the-road) task time and eyes-off-the-road time are correlated with static task time, the time to complete the task when the vehicle is parked (e.g., Green, 1999d; Farber et al., 2000; Young et al., 2005). Static task time is easy to measure and can be done using prototypes available early in design.

The current SAE J2364 test procedure (Society of Automotive Engineers, 2004a) requires that 10 subjects between the ages 45 and 65 be tested. Each subject completes five practice trials and three test trials in a *parked* vehicle, simulator, or laboratory mockup. The test cases to be examined (addresses for destination entry) are to be representative of what is planned for production.

In the static method, the subject performs the task, with the duration being from when the subject is told to start until the goal is achieved. Timing is continuous except for computational interruptions equal to or greater than 1.5 seconds, a time period during which the device is computing (for example, a route). If feedback is provided to the driver, that period is excluded from the 15-second task time limit. The interface complies with J2364 if the mean of the log of the task times is less than the log of 15 seconds. (Logs were used to reduce the influence of long outliers.)

It must be emphasized that the 15-second limit is for a static test. On-the-road drivers will take 30% to 50% longer overall, and furthermore alternate between looking inside the vehicle and looking at the road. The test procedure does not suggest drivers can safely look away from the road continuously for 15 seconds.

Some have argued that use of static task time fails to identify interfaces requiring long glance durations. However, analysis of real products shows the primary risk is from tasks that take too long to complete. In fact, it is very difficult to think of driver tasks for navigation systems that have short total task times but very long glance durations. In real driving, people truncate glances to the interior when the glances become too long but tend to complete tasks, even if they are unacceptably long. In practice, eliminating tasks with long completion tasks (the worst tasks) also eliminates many of the tasks with long glance durations.

Nonetheless, J2364 provides an alternative method involving visual occlusion. In that method, either the subjects wear special LCD glasses, or vision to the device is otherwise periodically interrupted, simulating looking back and forth between the road and the device. (Unlike driving, though, subjects do nothing in the occlusion interval.) The device is visible for 1.5 seconds and occluded for 1 to 2 seconds, with 1.5 seconds being recommended. Compliance is achieved if the sum of the log of the viewing times is less than the log of 20 seconds.

Alliance Principles

The Alliance of Automobile Manufacturers, the trade association of most major manufactures of automobiles in the United States (except for Nissan, Honda, Subaru, and Isuzu), has devoted considerable effort in developing design guidelines. Although the document scope states that it applies only to advanced information and communication systems, it should apply to traditional information or communications systems (e.g., entertainment systems), since the low-level tasks (reading displays, pressing buttons, etc.) and the manner in which those tasks interfere with driving are the same for all systems.

Their design principles document is a detailed elaboration of the 24 principles in the EU guidelines (e.g., "The system should be located . . . in accordance with relevant . . . standards. . . ." "No part . . . should obstruct any vehicle controls or displays. . ."), which seem obvious at a high level; however, the important content of the principles is in how they are achieved. Each principle has four parts: (a) rationale, (b) criterion/criteria, (c) verification procedures, and (d) examples.

The most important principle is 2.1: "Systems with visual displays should be designed such that the driver can complete the desired task with sequential glances that are brief enough not to adversely affect driving." Two alternative sets of criteria are offered. Alternative A says that the 85th percentile of single-glance durations should generally not exceed 2 seconds, and task completion should not require more than 20 seconds of total glance time. Verification can be achieved by a visual occlusion procedure (1.5 second viewing time, 1.0 second occlusion time), or by monitoring eye fixations directly using either a camera aimed at the face or an eye fixation monitoring system in either a divided-attention or on-road test.

Alternative B requires that the number of lane exceedances do not exceed the number associated with a reference task such as manual radio tuning, and that cars following headway should not degrade under those conditions, either. The verification procedure is stipulated to be driving on a divided road (either real or simulated) at 45 mph or less in daylight, on dry pavement, with low to moderate traffic. Additional details are provided describing the location of the radio, the stations to choose among, what constitutes a trial, subject selection (equal numbers of men and women between the ages of 45 and 65), and so forth.

Although both procedures seem well described on the surface, additional details and constraints are needed to make those procedures repeatable. For example, the differences in performance between driving in "low" and "medium" traffic could be quite considerable. One source of ideas is Lai's (2005) dissertation, which specifically proposes using variations in lateral position and variations in speed maintenance as assessment measures. Nonetheless, the principles represent a reasonable first step in developing a test protocol. On the other hand, on-road and simulator tests are extremely expensive and time-consuming, often impractical, and occur too late to have a useful impact. It is therefore the author's preference to rely on

simpler evaluation procedures such as J2364 and the J2365 calculation procedure described in the next section.

WHAT TOOLS AND ESTIMATION PROCEDURES CAN AID TELEMATICS DESIGN?

SAE J2365 Calculations

SAE J2365 (Green, 1999a) was developed to allow designers and engineers to calculate completion times early in design, when it is still a concept that can easily be modified. As with J2364, J2365 is for in-vehicle tasks involving visual displays and manual controls evaluated statically—that is, while parked (or in a bench-top simulation). SAE J2365 applies to both OEM and aftermarket equipment. Though intended for navigation systems, J2365 should provide reasonable estimates for most in-vehicle tasks involving manual controls and visual displays.

The calculation method is based on the Goals, Operators, Methods, and Selection rules (GOMS) model described by Card, Moran, and Newell (1980) with task time data from several sources. (See Kieras' chapter 60 for background information on GOMS.) The keystroke data was drawn from UMTRI studies of the Siemens Ali-Scout navigation system (Steinfeld, Manes, Green, & Hunter, 1996; Manes, Green, & Hunter, 1998). Search times were based on Olson and Nilsen (1987–1988) and the mental time estimates were drawn from the Keystroke-Level Model (Card, Moran, and Newell, 1983) and UMTRI Ali-Scout studies. Thus, the times shown in Table 35.13 have been tailored for the automotive context. (See also Nowakowski, Utsui, & Green, 2000.)

The basic approach involves top-down, successive decomposition of a task. The analyst divides the task into logical steps. For each step, the analyst identifies the human and device task operators. Sometimes analysts get stuck using this approach because they are not sure how to divide a task into steps. In those cases, utilizing a bottom-up approach may overcome such roadblocks. For each goal, the analyst identifies the method used.

The analyst is advised to use paragraph descriptions of each method to document them and then convert those descriptions into pseudo code. All steps are assumed to occur in series; multiple tasks cannot be completed at the same time. Furthermore, most drivers are assumed to use only visible, noncognitively loading shortcuts. Invisible shortcuts are likely to be used only by experts.

Next, the pseudo code task description is entered into an Excel spreadsheet. The analyst looks up the associated time for each operator listed in Table 35.10 and sums them to determine total task time. To assist in understanding the process, the practice provides a step-by-step example of entering a street address into a PathMaster/NeverLost navigation system, a popular U.S. product. For background on the calculation method, see Green (1999b).

The J2365 approach shares a number of assumptions, many of which are also shared with the basic GOMS model. For example, the model assumes error-free performance, which while not likely, can be adjusted for (say by increasing the computed value by 25%). Further, activities are assumed to be routine cognitive tasks, with users knowing each step and executing them in a serial manner. Again, adjustments in computed time can account for users sometimes forgetting what is next.

Though many of these assumptions are not true, adjustments can be made for them and many times the adjustments are small. Furthermore, violations of assumptions tend to affect all interfaces equally, so decisions about which of several interfaces is best still hold. As a practical matter, the estimates are good enough for most engineering decisions. Readers should keep in mind that J2364 only requires the use of 10 subjects at most, so there is some error in those estimates. Those errors are likely to be as large as variability among analysts and among J2365 estimates.

IVIS Estimates

A more complex estimation procedure, the IVIS DEMAnD model (In-Vehicle Information System Design Evaluation and Model

TABLE 35.13 Operator Times (seconds)

			Time (s)	
Code	Name	Operator Description	Young Drivers (18–30)	Older Drivers (55–60)
Rn	Reach near	From steering wheel to other parts of the wheel, stalks, or pods	0.31	0.53
Rf	Reach far	From steering wheel to center console	0.45	0.77
C1	Cursor once	Press a cursor key once	0.80	1.36
C2	Cursor 2 times or more	Time/keystroke for the second and each successive cursor keystroke	0.40	0.68
L1	Letter or space 1	Press a letter or space key once	1.00	1.70
L2	Letter or space 2 times or more	Time/keystroke for the second and each successive cursor keystroke	0.50	0.85
N1	Number once	Press the letter or space key once	0.90	1.53
N2	Number 2 times or more	Time/keystroke for the second and each successive number key	0.45	0.77
E	Enter	Press the Enter key	1.20	2.04
F	Function keys or shift	Press the function keys or Shift	1.20	2.04
M	Mental	Time/mental operation	1.50	2.55
S	Search	Search for something on the display	2.30	3.91
Rs	Response time of system-scroll	Time to scroll one line	0.00	0.00
Rm	Response time of system-new menu	Time for new menu to appear	0.50	0.50

Note 1: The keystroke times shown in Table 35.12 include the time to move between keys.
Note 2: System response times to display new menus may be empirically determined.

of Attention Demand, described by Hankey, Dingus, Hanowski, Wierwille, & Andrews, 2000a, 2000b). The model, which runs under Windows 98 or Windows NT, allows analysts to calculate a wide range of performance characteristics for proposed user interfaces. (The CD can be obtained from the U.S. Department of Transportation, Federal Highway Administration, Turner-Fairbank Highway Research Center in McLean, Virginia.)

Consistent with common understanding, the model assumes there are five basic human resources: (a) visual input, (b) auditory input, (c) central processing, (d) manual output, and (e) speech output. Overload of any one of these resources will affect task performance. The more demanding an in-vehicle task, the greater the likelihood of a crash. The model does not include a haptic component because haptic displays are rare in contemporary vehicles. Although many have developed models of human performance that partition human resources more finely, a five-component model is sufficient for most in-vehicle analyses. The data in the model were based on four experiments: (a) Gallagher (2001), (b) Blanco (1999), (c) Biever (1999), and (d) research conducted by Westat, a consulting company. These experiments concentrated on reading visual displays while driving, though there was work on auditory information as well.

To use the model, analysts need to create a description of each task drivers perform. Generally, that involves selecting the task in question from a large library of tasks in the database, modifying an existing description, or creating one from scratch. Tasks are grouped into seven categories: (a) conventional, (b) search, (c) search-plan, (d) search-plan-interpret, (e) search-plan-compute, (f) search-compute, and (g) search-plan-interpret-compute. Analysts need to select the driver age category (or specify all ages), the traffic density, the road complexity, the reliance on symbols/labels, the location of the display, and other characteristics.

The output of the model includes about 20 parameters, such as (a) the expected number of glances, (b) total task time, (c) ratings of mental demand and frustration, (d) total task demand, and so forth. In addition, the model output specifies if design thresholds are exceeded. The model proposes two sets of thresholds: (a) yellow-line and (b) red-line. Yellow-line thresholds were sets of points at which there was a measurable degradation in driving performance ($p < .05$) from baseline driving in the research conducted to support model development. Red-line thresholds indicated that a composite group of surrogate safety measures of driving performance was substantially affected. The red-line values were determined primarily from the literature and expert opinion. Table 35.14 shows those thresholds.

Independent validation data of the model has yet to appear in the literature.

TABLE 35.14. IVIS Yellow and Red Line Thresholds from Hankey, Dingus, Hanowski, Wierwille, and Andrews (2000a)

Measure	Affected (yellow)	Substantially Affected (red)
Single glance time	1.6 s	2.0 s
Number of glances	6 glances	10 glances
Total visual task time	7 s	15 s
Total task time	12 s	25 s

CLOSING THOUGHTS

This chapter makes the following key points:

1. Driving is quite different from sitting at a desk in an office because of the concern for crash risk.
2. A large number of new telematics applications will appear in vehicles over the next few years. They should radically reshape the driver's task, providing the driver with a flood of information.
3. This flood of information has the potential to distract drivers from driving. Distraction is associated with specific types of crashes (single vehicle run off road, rear end) that are most common under generally good driving conditions.
4. Relative to other distractions, the number of telematics-related crashes has been low in the recent past; however, current projections are that in excess of 200 people are killed per year in the U.S. in cell-phone-related crashes, and the number is growing. This number of fatalities exceeds that from front air bags and Firestone/Explorer tire rollovers, both of which had major financial impact on industry and led to federal regulatory changes.
5. Workload managers may be a long-term alternative to legislation to reduce distraction that leads to crashes.
6. Trips are made for a wide variety of purposes that need to be considered in assessing telematics applications.
7. Telematics need to be developed for a wide range of driver ages. Even well into their retirement years, most people still drive and will need to be able to use telematics.
8. Though people commonly think of passenger cars as the primary form of personal transportation in the United States, more trucks (especially pickups, SUVs, and vans) are produced in the United States than passenger cars, and they will be the most common platform in the future.
9. In assessing safety and usability, a wide variety of measures of longitudinal and lateral control are used in addition to task completion time, errors, and subjective ratings of ease of use. A major challenge in interface evaluation is dealing with performance trade-offs between measures.
10. Over the last few years, there have been significant advances in driving simulators and instrumented vehicles that have improved their quality and reduced their cost for safety and usability evaluations; however, the cost of these systems may be out of the range of most laboratories, especially those in academic settings.
11. The Alliance, Battelle, EU, HARDIE, JAMA, SAE, TRL, and UMTRI, along with ISO, have developed key design documents. The SAE recommended practices (especially SAE J2364, the 15-second rule) and the Alliance guidelines have the most impact in the United States, as does JAMA in Japan. ISO is in the process of developing a large set of standards and other organizations are updating theirs.
12. SAE J2365 and IVIS can assist in predicting task performance time, a simple measure for evaluating telematics safety and usability.

Thus, while the HCI literature provides a framework for test methods and evaluation, a great deal is specific to the motor

vehicle context because of the safety-critical nature of the context and the timesharing not found in office activities. To meet the needs of the future, the cost of the methods needs to be reduced, and reliable tools, especially for recording eye fixations, are needed. Significant research is needed to support the development driver performance models (and workload managers) and understand how drivers use real telematics applications.

References

Alliance of Automobile Manufacturers. (2003, June 17). Statement of principles on human-machine interfaces (HMI) for in-vehicle information and communication systems (draft). Washington, DC: Author. http://www.umich.edu/~driving/guidelines/guidelines.html).

Barfield, W., & Dingus, T. A. (1997). *Human factors in intelligent transportation systems*. Mahwah, N. J.: Lawrence Erlbaum Associates.

Biever, W. J. (1999). *Auditory based supplemental information processing demand effects on driving performance*. Unpublished master's thesis, Virginia Polytechnic Institute and State University, Blacksburg.

Blanco, M. (1999). *Effects of in-vehicle information systems (IVIS) tasks on the information processing demands of a commercial vehicle operations (CVO) driver*. Unpublished master's thesis, Virginia Polytechnic Institute and State University, Blacksburg.

Campbell, J. L., Carney, C., & Kantowitz, B. H. (1997). *Human factors design guidelines for advanced traveler information systems (ATIS) and commercial vehicle operations* (CVO; Technical Report FHWA-RD-98-057). Washington, DC: U.S. Department of Transportation, Federal Highway Administration.

Card, S. K., Moran, T. P., & Newell, A. (1980, July 23). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, (7),* 396–410.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Commission of the European Communities. (1999). Statement of principles on human machine interface (HMI) for in-vehicle information and communication systems. (Annex 1 to Commission Recommendation of 21 December 1999 on safe and efficient in-vehicle information and communication systems: A European statement of principles on human machine interface). Brussels, Belgium: European Union.

Crundall, D., Bains, M., Chapman, P., & Underwood, G. (2005). Regulating conversation during driving: a problem for mobile phones? *Transportation Research Part F: Traffic Psychology and Behaviour*, *8,* 197–211. Elsevier: Amsterdam.

Drews, F. A., Pasupathi, M., & Strayer, D. L. (2004). Passenger and cell-phone conversations in simulated driving. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting,* 2210–2212.

Driving Assessment 2005: Proceedings of the Third International Symposium on Human Factors in Driver Assessment, Training and Vehicle Design. (2005). Iowa City, Iowa: University of Iowa. Retrieved April 3, 2007, from http://ppc.uiowa.edu/driving-assessment/2005/final/index.htm

Eby, D. W., & Kostyniuk, L. P. (2004). *Distracted driving scenarios: A synthesis of literature, 2001 Crashworthiness Data System (CDS) data, and expert feedback* (SAVE-IT project technical report, Task 1). Washington, DC: U.S. Department of Transportation. Retrieved April 3, 2007, from (http://www.volpe.dot.gov/hf/roadway/saveit/docs/dec04/finalrep_1.pdf).

End-of-Year 2006 Top-Line Survey Results. (2006). Cellular Telecommunications and Inernet Association, Washington, D.C. Retrieved April 5, 2007, from http://files. ctia.org/pdf/CTIA_Survey_Year_End_2006_Graphics.pdf

Eoh, H., Green, P. A., Schweitzer, J., & Hegedus, E. (2006). *Driving Performance Analysis of the ACAS FOT Data and Recommendations for a Driving Workload Manager.* (Technical Report UMTRI-2006-18). Ann Arbor, Michigan: University of Michigan Transportation Research Institute.

Farber, E., Blanco, M., Foley, J. P., Curry, R., Greenberg, J., & Serafin, C. (2000), Surrogate measures of visual demand while driving. In *Proceedings of the IEA/HFES 2000 Congress* [CD-ROM]. Santa Monica, CA: Human Factors and Ergonomics Society.

Fatality Analysis Reporting System (FARS) Web-Based Encyclopedia. (2007). National Center for Statistics & Analysis, U.S. Department of Transportation. Retrieved April 3, 2007, from http://www-fars.nhtsa.dot.gov/

Gallagher, J. P. (2001). *An assessment of the attention demand associated with the processing of information for in-vehicle information systems (IVIS)*, Unpublished master's thesis, Virginia Polytechnic Institute and State University, Blacksburg.

Gould, J. D., & Lewis, C. (1985, March). Designing for usability: Key principles and what designers think. *Communications of the ACM, 28*(3), 300–311.

Green, P. (1995a). Automotive techniques. In Weimer, J., (Ed.), *Research techniques in human engineering* (2nd ed., 165–208). New York: Prentice-Hall.

Green, P. (1995b). *Measures and methods used to assess the safety and usability of driver information systems* (Technical report FHWA-RD-94-088). McLean, VA: U.S. Department of Transportation, Federal Highway Administration.

Green, P. (1995c). *Suggested procedures and acceptance limits for assessing the safety and ease of use of driver information systems* (Technical report FHWA-RD-94-089). McLean, VA: U.S. Department of Transportation, Federal Highway Administration.

Green, P. (1999a). Estimating compliance with the 15-second rule for driver-interface usability and safety. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting* [CD-ROM]. Santa Monica, CA: Human Factors and Ergonomics Society.

Green, P. (1999b). *Navigation system data entry: Estimation of task times* (Technical report UMTRI-99-17). Ann Arbor, MI: University of Michigan, Transportation Research Institute.

Green, P. (1999c). The 15-second rule for driver information systems. In *ITS America Ninth Annual Meeting Conference Proceedings* [CD-ROM]. Washington, DC: Intelligent Transportation Society of America.

Green, P. (1999d). *Visual and task demands of driver information systems* (Technical report UMTRI-98-16). Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. (2000a). *The human interface for ITS display and control systems: developing international standards to promote safety and usability*. Paper presented at the International Workshop on ITS Human Interface, Utsu, Japan. June 8, 2000.

Green, P. (2000b). Crashes induced by driver information systems and what can be done to reduce them (SAE paper 2000-01-C008). In *Convergence 2000 Conference Proceedings* (27–36), Warrendale, PA: Society of Automotive Engineers.

Green, P. (2001a). *Safeguards for on-board wireless communications*. Paper presented at the Second Annual Plastics in Automotive Safety Conference, Troy, Michigan. February 5, 2001.

Green, P. (2001b). *Synopsis of driver interface standards and guidelines for telematics as of mid-2001* (Technical report UMTRI-2001-23). Ann Arbor, MI: University of Michigan, Transportation Research Institute.

Green, P. (2001c). *Telematics: Promise, potential, and risks*. Presented as the Management Briefing Seminar panel session—Traverse City Conference. Ann Arbor, MI: University of Michigan, Transportation Research Institute, Office for the Study of Automotive Transportation. August 6, 2001.

Green, P. (2001d, February 19–20). *Variations in task performance between younger and older drivers: UMTRI research on telematics*. Paper presented at the Association for the Advancement of Automotive Medicine Conference on Aging and Driving, Southfield, MI.

Green, P. (2004). Driver distraction, telematics design, and workload managers: safety issues and solutions (SAE paper 2004-21-0022). In *Proceedings of the 2004 International Congress on Transportation Electronics* (pp. 165–180). Warrendale, PA: Society of Automotive Engineers.

Green, P. (2005). How driving simulator data quality can be improved. *Proceedings of the Driving Simulation Conference North America 2005 [CD-ROM].* November, 2005, Orlando, Florida. Retrieved April 3, 2007, from http://www.umich.edu/~driving/publications.html

Green, P., Flynn, M., Vanderhagen, G., Ziomek, J., Ullman, E., & Mayer, K. (2001). *Automotive industry of trends in electronics: year 2000 survey of senior executives* (Technical Report UMTRI-2001-15). Ann Arbor, MI, University of Michigan, Transportation Research Institute.

Green, P., Levison, W., Paelke, G., & Serafin, C. (1995). *Preliminary human factors guidelines for driver information systems* (Technical report FHWA-RD-94-087). McLean, VA: U.S. Department of Transportation, Federal Highway Administration.

Green, P., Nowakowski, C., Mayer, K., & Tsimhoni, O. (2003). Audiovisual system design recommendations from experience with the UMTRI driving simulator. *Proceedings of the Driving Simulator Conference North America 2003 [CD-ROM].* October 2003, Dearborn, Michigan. Retrieved April 3, 2007, from http://www.umich.edu/~driving/publications.html

Hankey, J. M., Dingus, T. A., Hanowski, R. J., Wierwille, W. W., & Andrews, C. (2000a). *In-vehicle information systems behavioral model and design support final report* (Technical report FHWA-RD-00-135). McClean, VA: U.S. Department of Transportation, Federal Highway Administration. Retrieved April 3, 2007, from (http://www.tfhrc.gov/humanfac/00-135.pdf).

Hankey, J. M., Dingus, T. A., Hanowski, R. J., Wierwille, W. W., & Andrews, C. (2000b). *In-vehicle information systems behavioral model and design support: IVIS DEMAnD prototype software user's manual* (Technical report FHWA-RD-00-136). McClean, VA: U.S. Department of Transportation, Federal Highway Administration. Retrieved April 3, 2007, from http://www.tfhrc.gov/humanfac/00-136.pdf

Highway Statistics 2003. (2004a). Section III: Driver Licensing. Office of Highway Policy Information, Federal Highway Administration: Washington, D.C. Retrieved April 3, 2007, from http://www.fhwa.dot.gov/policy/ohim/hs03/dl.htm.

Highway Statistics 2003. (2004b). Section II: Motor Vehicles. Office of Highway Policy Information, Federal Highway Administration: Washington, D. C. Retrieved April 3, 2007, from http://www.fhwa.dot.gov/policy/ohim/hs03/htm/mv1.htm.

Hoedemaeker, M., de Ridder, S. N., & Janssen, W. H. (2002). *Review of European human factors research on adaptive interface technologies for automobiles* (TNO report TM-02-C031). Soesterberg, The Netherlands: TNO Institute for Perception.

Horrey, W. J., & Wickens, C. D. (2004). Cell phones and driving performance: A meta-analysis. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (2304–2308). Santa Monica, CA: Human Factors and Ergonomics Society.

Hu, P. S., & Reuscher, T. R. (2004). Summary of Travel Trends: 2001 National Household Travel Survey. U. S. Department of Transportation: Washington, D. C. Retrieved April 3, 2007, from http://nhts.ornl.gov/2001/pub/STT.pdf

Insurance Institute for Highway Safety (IIHS). (2006). *US driver licensing procedures for older drivers.* Arlington, Virginia. Retrieved April 3, 2007, from http://www.iihs.org/laws/state_laws/older_drivers.html

International Organization of Standardization. (2003). *How are ISO standards developed?* Geneva, Switzerland. Retrieved April 3, 2007, from http://www.iso.ch/iso/en/stdsdevelopment/whowhenhow/how.html

International Organization of Standardization. (2007). *List of technical committees: TC 22/SC 13 Ergonomics applicable to road vehicles.* Geneva, Switzerland. Retrieved April 4, 2007, from http://www.iso.org/iso/en/CatalogueListPage.CatalogueList?COMMID=869&scopelist=PROGRAMME.

International Organization of Standardization. (2001). *Road vehicles—Measurement of driver visual behaviour with respect to transport information and control systems—Part 2: Equipment and procedures* (ISO Technical Specification 15007-2: 2001). Geneva, Switzerland: Author.

International Organization of Standardization. (2002a). *Road vehicles—Ergonomic aspects of transport information and control systems—Dialogue management principles and compliance procedures* (ISO Standard ISO 15005:2002). Geneva, Switzerland: Author.

International Organization of Standardization. (2002b). *Road vehicles—Measurement of driver visual behaviour with respect to transport information and control systems—Part 1: Definitions and parameters* (ISO Standard 15007-1: 2002). Geneva, Switzerland: Author.

International Organization of Standardization. (2003a). *Road vehicles—Ergonomic aspects of transport information and control systems—Procedure for assessing suitability for use while driving* (ISO Standard 17287:2003). Geneva, Switzerland: Author.

International Organization of Standardization. (2003b). Road vehicles—Ergonomic aspects of transport information and control systems—Specifications and compliance procedures for in-vehicle visual presentation (ISO Standard 15008: 2003). Geneva, Switzerland: Author.

International Organization of Standardization. (2004a). *Road vehicles—Ergonomic aspects of transport information and control systems (TICS)—Procedures for determining priority of on-board messages presented to drivers* (ISO Trial Standard 16951: 2004). Geneva, Switzerland: Author.

IInternational Organization of Standardization. (2004b). *Road vehicles—Ergonomic aspects of transport information and control systems—Specifications and compliance procedures for in-vehicle auditory presentation* (ISO Standard 15006: 2004). Geneva, Switzerland: Author.

International Organization of Standardization. (2005). *Road vehicles—Ergonomic aspects of in-vehicle presentation for transport information and controls systems—Warning systems* (ISO Technical Report 16352: 2005). Geneva, Switzerland: Author.

International Organization of Standardization. (2006). *Road vehicles—Ergonomic aspects of transport information and control systems—Occlusion method to assess visual distraction due to the use of in-vehicle information and communication systems* (ISO Draft Standard 16673). Geneva, Switzerland: Author.

Japan Automobile Manufacturers Association. (2004). *JAMA guideline for in-vehicle display systems, version 3.0*. Tokyo, Japan: Author. Retrieved April 3, 2007, from http://www.jama.or.jp/safe/guideline/pdf/jama_guideline_v30_en.pdf.

Johansson, E., Engstrom, J., Cherri, C., Nodari, E., Toffetti, A., Schindhelm, R., et al. (2004). *Review of existing techniques and metrics for IVIS and ADAS assessment* (AIDE deliverable 2.2.1). Brussels, Belgium: European Union.

Kayl, K. (2000). The networked car: Where the rubber meets the road. Retrieved April 5, 2007, from http://sun.systemnews.com/articles/32/1/ja/2746.

Koushki, P. A., Ali, S. Y., & Al-Saleh, O. I. (1999). Driving and using mobile phones: Impacts on road accidents. *Transportation Research Record*, 1694, 27–33. National Research Council, Washington, D.C.

Lai, F. C. H. (2005). *Driver attentional demand to dual task performance*. Unpublished doctoral dissertation, University of Leeds, Institute for Transport Studies, United Kingdom.

Manes, D., Green, P., & Hunter, D. (1998). *Prediction of destination entry and retrieval times using keystroke-level models* (Technical report UMTRI–96–37). Ann Arbor, MI: University of Michigan, Transportation Research Institute.

Mayhew, D. R., Simpson, H. M., & Pak, A. (2003). Changes in collision rates among novice drivers during the first months of driving. *Accident Analysis and Prevention, 35*(5), 683–691. Elsevier: Amsterdam.

McEvoy, S. P, Stevenson, M. R., McCartt, A. T., Woodward, M., Haworth, C., Palamara, P., et al. (2005, July 12). Role of mobile phones in motor vehicle crashes resulting in hospital attendance: A case-crossover study. *British Medical Journal,* 1–5. Retrieved April 3, 2007, from http://bmj.bmjjournals.com/cgi/content/abstract/bmj.38537.397512.55v1

Michon, J. A. (Ed.) (1993). *Generic intelligent driver support*. London: Taylor & Francis.

Murray, C. J. L., & Lopez, A. D. (1996b). *The global burden of disease*. Boston: Harvard School of Public Health.

National Safety Council. (2004). *Injury facts*. Itasca, IL: Author.

Nowakowski, C. Friedman, D., & Green, P. (2001). *Cell phone ring suppression and HUD caller ID: Effectiveness in reducing momentary driver distraction under varying workload levels* (Technical report 2001-29). Ann Arbor, MI: University of Michigan, Transportation Research Institute. Retrieved April 3, 2007, from (http://www.umich.edu/~driving/publications.html).

Nowakowski, C., & Green, P. (2000). *Prediction of menu selection times parked and while driving using the SAE J2365 method* (Technical report 2000-49). Ann Arbor, MI, University of Michigan, Transportation Research Institute.

Nowakowski, C., Utsui, Y., & Green, P. (2000). *Navigation system evaluation: the effects of driver workload and input devices on destination entry time and driving performance and their implications to the SAE recommended practice* (Technical report UMTRI-2000-20). Ann Arbor, MI, University of Michigan, Transportation Research Institute.

Noy, Y. I. (Ed.). (1997). *Ergonomics and safety of intelligent driver interfaces*. Mahwah, NJ: Lawrence Erlbaum Associates.

Olson, J. R., & Nilsen, E. (1988). Analysis of the cognition involved in spreadsheet software interaction. *Human-Computer Interaction, 3*, 309–349.

Parkes, A. M., & Franzen, S. (1993). *Driving future vehicles*. London: Taylor and Francis.

Peacock, B., & Karwowski, W. (1993). *Automotive ergonomics*. London: Taylor & Francis.

Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., et al. (Eds.). (2004). *World report on road traffic injury prevention*. Geneva, Switzerland: World Health Organization. Retrieved April 3, 2007, from http://www.who.int/world-health-day/2004/infomaterials/world_report/en/

Pena, P. (2007). *An Education on Common Objections to Cell Phone Legislation.* Retrieved April 3, 2007,. from http://www.geocities.com/morganleepena/rebuttal.htm

Piechulla, W., Mayser, C., Gehrke, H., & König, W. (2003). Reducing Drivers' mental workload by means of an adaptive man-machine interface. *Transportation Research Part F: Traffic Psychology and Behaviour*, *6,* 233–248. Elsevier: Amsterdam.

Quimby, A. (1999). *A safety checklist for the assessment of in-vehicle information systems: scoring performance* (Project Report PA3536-A/99). Crowthorne, UK: Transport Research Laboratory.

Redelmeier, D. A., & Tibshirani, R. J. (1997). Association between cellular-telephone calls and motor vehicle collisions. *New England Journal of Medicine, 336*, 453–458.

Redelmeier, D. A., & Tibshirani, R. J. (2001). Car phones and car crashes: some popular misconceptions. *Canadian Medical Association Journal, 164*(11), 1581–1582.

Ribbens, W. B., & Cole, D. E. (1989). *Automotive electronics Delphi*. Ann Arbor, MI: Office for the Study of Automotive Transportation, University of Michigan, Transportation Research Institute.

Richardson, B., & Green, P. (2000). *Trends in North American intelligent transportation systems: A year 2000 appraisal* (Technical Report UMTRI–2000-9). Ann Arbor, MI, University of Michigan, Transportation Research Institute.

Roskam, A. J., Brookhuis, K. A., deWaard, D., Carsten, O. M. J., Read, L., Jamson, S., et al. (2002). *Development of experimental protocol* (HASTE Deliverable 1). Brussels, Belgium: European Commission.

Ross, T., Midtland, K., Fuchs, M., Pauzie, A., Engert, A., Duncan, B., et al. (1996). *HARDIE design guidelines handbook: human factors guidelines for information presentation by ATT systems*. Luxembourg: Commission of the European Communities.

Schindhelm, R., Gelau, C., Keinath, A., Bengler, K., Kussmann, H., Kompfner, P., et al. (2004). Report on the review of available guidelines and standards (AIDE deliverable 4.3.1). Brussels, Belgium: European Commission. Retrieved April 4, 2007, from http://www.aide-eu.org/pdf/sp4_deliv/aide_d4-3-1.pdf

Sloss, D., & Green, P. (2000). National automotive center 21st century truck (21T) dual use safety focus (SAE Paper 2000-01-3426). Warrendale, PA: Society of Automotive Engineers (published in National Automotive Center Technical Review, Warren, Michigan, U.S. Army Tank-Automotive and Armaments Command, National Automotive Center, 63–70).

Society of Automotive Engineers. (2002, May), *Calculation of the time to complete in-vehicle navigation and route guidance tasks* (SAE recommended practice J2365). Warrendale, PA: Author.

Society of Automotive Engineers. (2004a, May). *Navigation and route guidance function accessibility while driving* (SAE recommended practice 2364). Warrendale, PA: Author.

Society of Automotive Engineers (2004b, May). *Rationale document for SAE J2364* (SAE information report J2678). Warrendale, PA: Author.

Society of Automotive Engineers. (2005). *SAE handbook 2005*. Warrendale, PA: Author.

Steinfeld, A., Manes, D., Green, P., & Hunter, D. (1996). *Destination entry and retrieval with the Ali-Scout navigation system* (Technical report UMTRI–96-30, also released as EECS-ITS LAB FT97-077). Ann Arbor, MI: University of Michigan, Transportation Research Institute.

Stutts, J. C., Reinfurt, D. W., Staplin, L., & Rodgman, E. A. (2001). *The role of driver distraction in traffic crashes*. Washington, DC: AAA Foundation for Traffic Safety. Retrieved April 3, 2007, from (http://www.aaafts.org/pdf/distraction.pdf).

Sundeen, M. (2005). *Cell Phones and Highway Safety: 2005 State Legislative Update.* National Council of State Legislatures. Denver, Colorado. Retrieved April 3, 2007, from http://www.ncsl.org/programs/transportation/cellphoneupdate05.htm

Tierney, C. (2005. Big 3 market share dips to all-time low. *The Detroit News*, January 5, 2005. MediaNews Group: Detroit, Michigan. Retrieved April 4, 2007, from http://www.detnews.com/2005/autosinsider/0501/06/A01-50668.htm

Tijerina, L., Angell, L., Austria, A., Tan, A., & Kochhar, D. (2003). *Driver workload metrics literature review*. Washington, DC, U.S. Department of Transportation, National Highway Traffic Safety Administration.

U.S. Department of Transportation. (2005a). *2004 Traffic Safety Annual Assessment—Early Results* (Technical Report DOT HS 809 897). Washington, DC: Author. Retrieved April 3, 2007, from http://www-nrd.nhtsa.dot.gov/pdf/nrd-30/NCSA/RNotes/2005/809_897/

U.S. Department of Transportation. (2005b). Driver Cell Phone Use in 2005—Overall Results (Technical Report DOT HS 809 967). Washington, D.C.: Author. Retrieved April 3, 2007, from http://www-nrd.nhtsa.dot.gov/pdf/nrd-30/NCSA/RNotes/2005/809967.pdf

Underwood, S. E. (1989). Summary of preliminary results from a Delphi survey on intelligent vehicle-highway systems (technical report). Ann Arbor, MI: University of Michigan.

Underwood, S. E. (1992). *Delphi forecast and analysis of intelligent vehicle-highway systems through 1991: Delphi II. Ann Arbor, Program in Intelligent Vehicle-Highway Systems* (IVHS Technical Report-92-17), Ann Arbor, MI: University of Michigan.

Underwood, S. E., Chen, D., & Ervin, R. D. (1991). Future of intelligent vehicle-highway Systems: A Delphi forecast of markets and socio-technological determinants. Transportation Research Record No. 1305, 291–304.

Violanti, J. M., & Marshall, J. R. (1996). Cellular phones and traffic accidents: An epidemiological approach. *Accident Analysis & Prevention*, 28, 265–270. Elsevier: Amsterdam.

Wang, J.-S., Knipling, R. R., & Goodman, M. J. (1996). The role of driver inattention in crashes: New statistics from the 1995 crashworthiness data system. *Association for the Advancement of Automotive Medicine 40th Annual Conference Proceedings* (377–392). Des Plaines, IL: Association for the Advancement of Automotive Medicine.

Wierwille, W. (1995). Development of an initial model relating driver in-vehicle visual demands to accident rate. *Proceedings of the Third Annual Mid-Atlantic Human Factors Conference* (1–7). Blacksburg, VA: Virginia Polytechnic Institute and State University.

Wooldridge, M., Bauer, K., Green, P., & Fitzpatrick, K. (2000). *Comparison of workload values obtained from test track, simulator, and on-road experiments*. Paper presented at the Transportation Research Board Annual Meeting, Washington, DC.

Young, R., Aryal, B., Muresan, M., Ding, Z., Oja, S., & Simpson, N. (2005). Road-to-Lab: Validation of the static load test for predicting on-road driving performance while using advanced in-vehicle information and communication devices. *Driving Assessment 2005: Proceedings of the Third International Symposium on Human Factors in Driver Assessment, Training and Vehicle Design,* 240–254. Iowa City, Iowa: University of Iowa. Retrieved April 3, 2007, from http://ppc.uiowa.edu/driving-assessment/2005/final/index.htm

·36·

HUMAN-COMPUTER INTERACTION IN AEROSPACE

Steven J. Landry
Purdue University

Human-Computer Interaction in Aerospace **722**
Past Challenges, Solutions, and Principles **722**
Single-Sensor Single-Instrument (SSSI) Displays versus Integrated Displays 722
Basic Display Principles 723
Check-reading display research 724
Qualitative instrument research 724
Arrangement of Displays 724
How Much Information Can Be Displayed Effectively? 725
Clutter 725
Displaying Spatial Relationships 725
Introducing Automation to the Flight Deck **726**
Autopilots 726
Role Allocation 727
Fitts list 727
The "automate everything" philosophy 727
Human supervisory control 728
Levels of automation 728
Ecological allocation 729
Guidelines for automation 729
Early Air-Traffic Control 730
Early radar displays 730
Flight strips and strip bays 730
Modern Challenges, Solutions, and Principles **730**
Glass Cockpits, Flight Management Systems, Mental Models, and Mode Confusion 730
Heads-Up Displays 731
Information Automation 732
Alerting systems 732
Decision support systems 732
Modern Air-Traffic Control 733
Automated flight strips and situation awareness 733
Decision support tools in air-traffic control 734
Future Challenges **735**
Three-Dimensional Displays 735
Tunnels in the sky 735
The scale problem in air-traffic control 735
Coordination Support Systems 735
Technologies Needed for 33 Air Traffic 736
Datalink and the loss of the party line 736
4D contracts: Future changes in the roles of pilots and controllers? 736
Automating separation assurance 736
UAVs in the Airspace System 737
Conclusions **737**
References **737**

HUMAN-COMPUTER INTERACTION IN AEROSPACE

Aviation's interest with human-computer interaction (HCI) concepts began with the need to understand how pilots could interact with mechanical displays that indicated such things as airspeed, altitude, aircraft orientation, heading, and bearing from a radio navigation aid. As aircraft technology progressed and displays became more numerous and more complex, the interface issues became more pronounced. As travel by aircraft was commercialized, the consequences of these interface issues became more severe. As modern air-traffic control was introduced, a new arena for human interface issues opened. By the 1970s, computers were in commercial aircrafts, and air-traffic control was utilizing radar displays and automated flight data processing. This progression has not stopped, as higher levels of automation and new types of interfaces have been steadily added to flight decks and air-traffic control rooms over the succeeding decades.

This continuous progression of displays and automation has taxed the capabilities of researchers to fully investigate the vast and varied challenges associated with aviation HCI. This chapter discusses the efforts that have been made, are being made, and those issues which are just now on the horizon.

PAST CHALLENGES, SOLUTIONS, AND PRINCIPLES

Human factors issues related to aviation displays were recognized as early as 1923, as indicated by the following quotation from a National Advisory Committee for Aeronautics (the predecessor to NASA):

> The reaction of the aviator to his instruments has to be considered, as well as the operation of the instruments themselves. This is evident enough in the case of appliances such as oxygen apparatus, intended solely for the comfort and efficiency of the aviator, or in the case of complicated instruments such as bombsights. It is equally true, however, with the more simple, direct-reading instruments. It is not enough for such instruments to be mechanically correct; they must be, in the case of service instruments, readily intelligible to the pilot. The manipulation of the instrument must not make an appreciable demand on his time or attention. The visibility must be satisfactory both day and night. Finally, service instruments must be "foolproof." While much can be accomplished by technical instruction courses for aviation personnel, still the personal prejudice of the average pilot has to be reckoned with. If the instrument for any reason fails to appeal to the individual pilot, he will take great chances rather than trouble to look at it. On the other hand, if the instrument pleases his fancy, he may grow so attached to it that he will claim he could not fly safely without it, even though the instrument is scientifically known to be incorrect. Curious examples of this circumstance were found in the popularity of the earlier liquid type Pitot tube among the British pilots and the spinning-top inclinometer among the French. (Hersey, 1923, pp. 8–9)

Another early example of aviation cognitive engineering research was in 1939, when an Applied Psychology Unit at Cambridge University in England (headed by Frederic Bartlett) began studying the design of aviation equipment (Bartlett, 1943). Bartlett's group studied manual control performance, the interrelationship between controls and displays, and human vigilance. Also in 1939, the National Research Council in the United States created a Council on Aviation Psychology. The U.S. Army and Navy created programs on aviation psychology in 1940 and 1945, respectively.

World War II provided a wealth of aviation experience, and after the war was over, aviation-savvy graduate students entering universities on the GI Bill. At the same time, advances were being made in navigation systems technology, cathode ray tube (CRT) displays, aircraft instrumentation, and control technology. The rapid increase in civil aviation made the development of air-traffic control a priority as well, and the advent of radar displays peaked interest in cognitive engineering issues for air-traffic controllers.

These developments led to an explosion of research in the 1950s, both in the United States and in Europe. One of the more influential reports was a 1951 blue ribbon committee report titled "Human Engineering for an Effective Air-Navigation and Traffic-Control System" (Fitts, 1951b). This paper summarized the state of research as it concerned air transportation, and identified key areas of future research. As such, it provides a convenient milepost for a discussion of the history of aviation cognitive engineering research.

This chapter will begin from a historical perspective, covering the major issues conceived of in the 1950s and how they were addressed in the succeeding decades, if at all. Following that will be a discussion of modern concerns, which grew out of the computerization of flight decks and innovative display concepts such as heads-up displays. Lastly will be a discussion of the future challenges for aviation and air-traffic control displays.

One concern for researchers in the 1950s regarded the display of information on "transitory displays," mainly dials and gauges. As such, the main focus was on (a) whether (and how) to integrate information on a single display, (b) how to arrange displays, (c) how displays, controls, and the comprehension of spatial information were related, and even (d) what types of displays were best for certain functions.

Many of these questions have been successfully addressed, providing today's engineers with some of the most concrete guidance that cognitive engineering has to offer. Some of them continue to be fascinating topics of research to this day, and of course, many new topics related to information display and assessment have arisen over the years.

Single-Sensor Single-Instrument (SSSI) Displays versus Integrated Displays

The problems inherent in SSSI displays have become increasingly obvious as the number and complexity of sensors (and other information sources) has increased. This has been particularly noticeable in aviation, where flight decks have undergone significant changes since 1951, but has also been true for nuclear plants, which were just beginning to be tested in 1951.

Interestingly, air-traffic controllers have only recently begun to encounter this problem as the number of displays (and elements on those displays) provided to a controller (and in the Traffic Management Unit [TMU]) has substantially increased.

SSSI displays have been gradually replaced (or incorporated into multifunction displays) as display technology has improved, although there are still examples of traditional SSSI displays on older aircraft and plants (it is an expensive venture to replace them with graphical displays). It seems clear from the research, however, that fewer displays are better than many displays (although the former is not without its own problems).

The reason that SSSI displays are generally considered to be worse than an integrated display seems to be related to the fact that a person's ability to monitor dynamic information is very low, even when only keeping track of a few items (Monty, 1973). Also, aircraft operators have demonstrated that they have difficulty integrating a large amount of displayed information (Murphy, McGee, Paler, Paulk, & Wempe, 1978). These findings mean that it is difficult for an operator to maintain comprehension of a dynamic situation when forced to scan from display to display. A study of the operation of manipulators showed subjects performed 30% to 40% better using two displays as opposed to four displays, despite a greater amount of information in the four-display case (Bejczy & Paine, 1977). This result extended a much earlier finding showing that fewer detections would be missed detections if there were, for example, five times the error rate on one display as compared to five displays each with one fifth the error rate (Conrad, 1951).

The presence of large numbers of displays also increases the amount of irrelevant data through which the operator must sort, leading to the problem of clutter. If the SSSI displays must, at every moment, display all the information that may be needed in any state of the system, then there must be irrelevant information present at any given time, and the operator must filter out this irrelevant information. As the quantity of irrelevant information increases, operator performance decreases, particularly if the information is similar (Hodge & Reid, 1971; Well, 1971; Dorris, Connolly, Sadosky, & Burroughs, 1977).

Irrelevant data is a problem, even assuming the operator knows what data is relevant (which is, and continues to be, its own problem). The relevant data must then be located from the different (and often, physically separated) instruments, and then mentally integrated to obtain the higher order representation. This operation can be cognitively intensive, and during situations of stress or short time constraints, it may not be possible (Vicente & Rasmussen, 1992).

In the next section, these issues—general design principles, arrangement of displays, and clutter—will be discussed in more detail. Many of the same (or similar) principles can be noticed in chapter 2 of this volume.

Basic Display Principles

In examining the question of whether fewer displays containing integrated information were better, and how information may be integrated, researchers have developed a number of principles for the design of displays. In particular, researchers early on studied problems in discerning the direction of motion of a pointer, and in the use of multipointer displays (which were being used in altimeter designs; Fitts, 1951a). Since then, a number of design principles with regard to such quantitative displays have been put forth (Sanders & McCormick, 1993).

One study compared digital speedometers to dials and curvilinear displays (all electronically generated). For accuracy and speed of reading, the digital display performed consistently better (Simmonds, Galer, & Baines, 1981). In general, digital displays have been found superior when a precise numeric value is required, and when the values presented remain visible long enough to be read.

Fixed-scale displays with moving pointers are better in the opposite cases—where precision is not required or the values change quickly. They are also better in cases where direction and rate of change of the values are important. For the design of analog displays, the following guidelines (Heglin, 1973) have been generally accepted:

- A moving pointer against a fixed scale is preferred.
- Small changes in quantity are more visible with a moving pointer display.
- If the quantity to be read is analogous to some physical interpretation, horizontal or vertical fixed scales with moving pointers should be used in order to provide a zero reference.
- Do not mix multiple moving-element indicators when they are related to the same function, as this can cause reversal errors.
- Moving scale displays can be used where the scale is too great to be displayed on the face of an instrument.
- If a control changes the quantity on the display, it is less ambiguous for the pointer to move than the scale.

In addition, for quantitative displays the numeric progression and length of scale units affect the speed and accuracy of quantitative reading (Whitehurst, 1982), and scale progressions should be in increments of 1s or 5s (to ease determination of readings in multiples of 1 and 5). Unusual progressions and decimals should be avoided. It was also found that the distance between graduation markers should be no less than 0.05 to 0.07 inches apart (greater under low illumination conditions), that scale markers should be used, and they should have graduation marks for the lowest unit to be read, but in no case lower than one fifth or one tenth (Cohen & Follert, 1970). Bar-type displays should only show one segment of the bar, should not have the bar extended to zero, and should not be used where a need for an indication at zero is needed (Green, 1984).

Altimeter designs were of particular concern in the late 1940s, due to a spate of accidents related to misreading them. One of the early altimeters responsible for some of the problems violated one of the principles above: it had three pointers on the same scale. The operator would have to read three pieces of information and integrate them to get altitude. In 1968, a study determined that an integrated vertical display of altitude was best in terms of time and accuracy of reading (Roscoe, 1968). In explaining the results, the researcher indicated that key factors were the analogy of the vertical instrument to altitude and the integration of all elements of altitude into one display.

The distinction between quantitative and check-reading instruments has survived to this day, with the added distinction of qualitative and situational awareness instruments (Sanders & McCormick, 1993). In fact, despite the prevalence of digital computers driving them, graphical displays often show representations of analog-like displays for instruments designed for qualitative or check-reading. This is because analog displays were found to be more quickly read for this purpose, and contributed less to workload, when compared to digital displays (Hanson, Payne, Shively, & Kantowitz, 1981).

Check-reading display research. Researchers demonstrated that for a number of instruments configured together for check reading, the pointers on round instruments were best positioned at 9 or 12 o'clock (Sanders & McCormick, 1993). When all pointers show nearly identical readings, the grouping of the instruments and positioning of the pointers appeal to the similarity gestalt of human cognition. Violation of the gestalt is immediately and reliably recognized. Further research indicated that the addition of a line joining the instruments at their null position improved or added to the gestalt and improved performance (Dashevsky, 1964).

Qualitative instrument research. The distinction does not appear to have been made between quantitative and qualitative (perhaps a subset of which is check reading) instruments before 1960. Qualitative instruments are distinct from quantitative in that the specific numerical value is irrelevant; instead the approximate value, its value in relation to some standard, its rate, or its trend are of interest. The difference may have first been noticed in a study that examined two types of readings of an instrument using three types of scales. In that study, subjects had to either read the quantitative value, or read high, low, or OK. Open-window displays (see Fig. 36.1) were read the quickest for quantitative values, but vertical scales were read quickest for the qualitative assessment (Elkin, 1959).

Qualitative scales also typically have some identification of ranges. This can be done through the use of color, or by the addition of coded markings that have some association to the meaning of the range (Sabeh, Jorve, & Vanderplas, 1958).

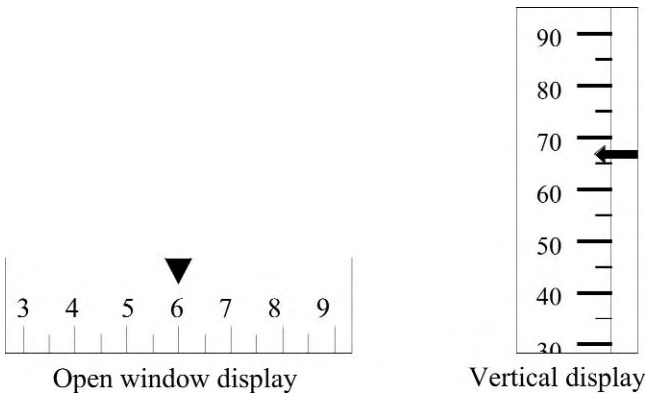


FIGURE 36.1. Display types.

Arrangement of Displays

A second question brought up in the Fitts (1951b) report regarded the organization of the display elements. A number of advances in relation to this arrangement problem were made shortly after its publication. Extensive studies (most notably one in which Fitts himself participated) were underway examining visual scanning patterns to determine the optimal layout of instruments (Fitts, Jones, & Milton, 1950). These studies led the Civil Aeronautics Board (later to become the FAA) to establish a standard arrangement for flight instruments, commonly referred to as the "basic T," in 1953 (Civil Aeronautics Board, 1953). It included six basic instruments, namely, (a) airspeed, (b) attitude, (c) altitude, (d) direction, (e) "climb" (which is taken here to mean vertical speed), (f) flight path deviation, and (g) their required arrangement. This was later reduced to just the first four instruments (Civil Aeronautics Board, 1957).

The analysis used in relating visual scanning patterns to display positioning was referred to as "link analysis." In this analysis, a "link" was sequential fixations on two items. The more occurrences of sequential fixations between two items, the stronger the link. Items strongly linked should then be located next to one another, a technique and principle used in other fields (such as in operation centers aboard naval vessels) (Chapanis, 1959). Other studies of eye movements determined that the best location for important information is just above or below the glareshield of a vehicle (Cole, Milton, & McIntosh, 1954). Since then, advances have been made in automating information organization (Mendel & Sheridan, 1986) and instrument-panel layout (Pulat & Ayoub, 1979), although methods of this sort are not widely used.

The layout of the instruments not only serves to reduce the time to scan the instruments, but can also affect workload. A principle of display design called the "proximity compatibility principle" states that, when attempting to integrate two sources of information from two locations, greater workload is induced by greater separation, as the physical proximity of the sources should mimic the cognitive proximity of the information (Wickens & Carswell, 1995).

This principle relates to the requirement of the operator to integrate different sources of information. There are two basic types of integration covered by this principle. The first involves locating a source of information (such as the heading indicator) next to related sources of information (such as bank indication and turn coordinator). The second involves the display itself integrating more than one piece of information, such as the attitude indicator showing both pitch and bank at the same time.

Multifunction graphical displays (MFDs) overcome these two limitations. These displays provide, in one viewport, a number of elements that may need to be integrated frequently (although not always), satisfying the proximity compatibility principle (Seidler & Wickens, 1992). The content of the MFD can be manually controlled, providing the opportunity to present a wealth of information for which there previously was no physical space.

Two problems arise, however. First, information that is not currently displayed is unavailable to the pilot. If that information changes, the pilot will be unaware of the change. Second, the "depth" added to the display presents the problem of navigation,

although this can be mitigated through good organization (and minimal depth) of the pages (Allen, 1983; Schneiderman, 1998, Francis, 2000). If these principles are not met, however, there is the possibility that the operator will become lost in the database. In addition, there is workload (and heads-down time) associated with the controls for the display.

How Much Information Can Be Displayed Effectively?

The problem of how much information can be fit on a display, has significantly evolved since 1951, but is still a topic of debate and research. A large body of work on visual processing has been amassed over the years, and many of the theories are still hotly debated. What is certain is that distinguishability of items on a display is overwhelmed by the somewhat limited perceptual processing capacity of the human. Without going into a great amount of detail, some of the findings with respect to this issue will be summarized.

There are two fundamental limits on extracting information from a visual scene: (a) processing and (b) attention. At any time, only a small amount of the information available to the retina can be processed (although all of it is perceived). The mechanics of processing are still being determined, but it appears likely that some high-level representation is quickly obtained, followed by a "weeding out" of some portions of the image (VanRullen & Thorpe, 2001). Scene processing allows a variety of information to be extracted, but specific objects of interest may or may not be extracted in sufficient detail. Focused attention captures this detail, but objects can only be processed serially, and subsequent processing of these objects is inhibited (referred to as "negative priming"; Tipper & Driver, 1988). The combination of these effects yields an effective capacity of 4 to 6 items (VanRullen & Koch, 2003). This number represents how many items can be represented in visual awareness at any one time. Objects/information may then be transferred to long-term memory, a process that has also been researched extensively.

How much information can be fit on one display, then, is dependent on the task and environmental factors. If one is reading a depiction of an instrument approach with the intention of memorizing details, and there is no time pressure, a great deal of information can be displayed and recalled effectively. If an air-traffic controller has a fraction of a second to glean track information from a radar display, he or she is unlikely to extract more than 4 to 6 pieces of information from that display. Moreover, what information would be extracted from that display is unclear, although factors of the displayed information (salience, size, color, etc.) can influence selective attention.

One further question remains in regards to information display that we will address here—the rate at which information can be extracted. This is also complex, dependent upon physiological factors, features of the display, complexity of the information to be extracted, and other factors. Simple responses to visual stimuli can occur within 30 to 60 msec (Iwasaki, 1996), but more complex objects (such as text) can require a great deal more time.

In summary, the answer to the question of how much information can be fit on a display is, "It depends." The question has become even more relevant as electromechanical displays have been almost universally replaced by computer-driven electronic displays, which can display a great deal of information in one place. These displays are also three-dimensional (3D) in the sense that they can contain pages of information (adding a new task—navigation of a display), and their contents can be modified by the user (or automation).

Clutter

Clutter is a significant issue in two main classes of displays: (a) multifunction displays (such as air-traffic control displays and flight deck displays that show route, weather, and traffic), and (b) heads-up displays (HUDs). The source of the clutter is different in the two cases. In the former case, a large quantity of data is being overlaid on the display, whereas in the latter, the display is overlaid on a source of great clutter (the outside world). In both cases, the problem is the same: response time for a stimulus is increased with the proximity of nonstimulus items (C. W. Eriksen & B. A. Eriksen, 1974). Additionally, clutter can increase the time it takes for the pilot to locate an item, or even obscure items of interest (Wickens, 2003). Clutter has also been mentioned by pilots in subjective ratings for some time (Abbott et al., 1980).

The solution to this problem is to provide the operator with decluttering capability—that is, the ability to either manually or automatically remove information from the display; however, if the information is no longer present, it cannot be integrated with other information, and changes to the information, which could be important, will of course not be noticed. This is in addition to the workload required to remove or add information to the display, which could be considerable, particularly in the case of navigating a multifunction display.

Displaying Spatial Relationships

The Fitts (1951b) report suggested that "studies should be made to determine principles governing the effective display of information about the relative position of aircraft and ground objects in tri-dimensional space" (p. xvii). Specifically, the committee felt that research should be directed at determining the best types of projections for displaying spatial relations, and how symbolic and pictorial displays are best used for this purpose. In addition, more studies were called for to determine what display characteristics were required when control manipulation resulted in changes in the display (e.g., turning a control yoke results in changes in bank and heading indications).

Aviation displays is a natural place for interest into the display of spatial relations to develop. Air-traffic controllers must gauge both absolute and relative position and motion of aircraft in at least two dimensions (often three), and provide spatial control guidance to pilots. A pilot must, through use of instrumentation, determine relative (and sometimes absolute) position in 3D coordinates, and keep track of 3D orientation. This is accomplished through the use of (a) an attitude indicator (ADI), which shows pitch and bank, (b) a heading indicator, which shows absolute orientation, (c) a turn indicator, which shows

yaw rate, and (d) one or more instruments showing relative position and bearing (e.g., altimeters, radio magnetic indicator, course deviation indicator, vertical deviation indicator, distance measuring equipment).

One problem that almost certainly drove the recommendation in the Fitts (1951b) report regarding the display of orientation was the problem of the ADI. In particular, what should move, the aircraft or the earth? Several principles related to this question (Roscoe, 1981) have been put forth in consideration of the body of research into interfaces:

- Principle of pictorial realism: The display should be pictorially analogous to the real world.
- Principle of compatible motion: The part that moves in the real world should be the part that moves on the display.
- Principle of integration: Related information should be integrated on the same display.
- Principle of pursuit presentation: A display should allow pursuit tracking versus just compensatory tracking—that is, if a target moves both absolutely and relatively, both sources of movement should be displayed.

So a moving aircraft on an ADI or an air-traffic controller's radarscope would be compatible with the principle of compatible motion; however, while the radarscope also complies with the principle of pictorial realism, from the pilot's perspective, the moving aircraft would (loosely) violate it; from the pilot's perspective, it appears that the horizon is banked as opposed to the aircraft. Although designers have settled on the moving earth model for ADI design, the debate was never satisfactorily resolved. Experiments based on (actually, an extension of) a proposal to integrate the two models (Fogel, 1959) showed improved performance (Beringer, Willeges, & Roscoe, 1975). This topic continues to be discussed to this day, with researchers arguing that the moving earth ADI has continued to be used only because of the lack of interest on the part of pilots to change it, despite evidence that it is has been causing accidents for decades (Bryan, Stonecipher, & Aron, 1954).

In addition to the orientation question addressed mainly by the ADI, there are questions about how best to portray relative position. This is a common problem for both air-traffic controllers, who must assess relative bearing and range between aircraft, and pilots, who must frequently assess their relative bearing and range from some navigational fix. More recently, the question of vertical navigation has received attention.

A typical horizontal-navigation problem for a pilot is common to anyone familiar with basic instrument flying. From any given position, a pilot could be asked to fly to a "fix," whose position is defined by bearing and range relative to a radio navigation aid (e.g., a VOR). Although this can be calculated from current heading and the ownship's bearing and range to the VOR, it is not a trivial task. A simpler task is to fly inbound on a course to a VOR or outbound on a radial from a VOR. Another example is the task of determining the position of the ownship given access to several radio navigational aids.

Recent research has qualified the difficulties inherent in these operations as "representational" problems (Zhang, & Norman, 1994). The theory follows from the argument that the information to accomplish many tasks is distributed between the external stimuli (the interface) and internal cognition (in this case, of the pilot; Norman, 1993). Furthermore, the information representations may have different scales: (a) ratio, (b) interval, (c) ordinal, or (d) nominal. Efficient external representations reflect all the categories of information present in the "real-world" object. Deficiencies in information would require that information to be distributed to the internal cognition of the operator, while surplus information (information present in the display that is not in the object) could be misleading.

Application of this theory to the problem of horizontal navigation shows that different equipment requires different levels of internal cognition. A simple ADF or VOR system requires internal computation of angular differences and range. A map type display, however, explicitly represents these, simplifying the task.

An extreme example of this representational problem is the difficulty in determining relative position from terrain (or weather) on a paper map (or even presented aurally!) from their knowledge of absolute position as given by the flight instruments. It is difficult and time-consuming to accurately comprehend this situation. This was apparently recognized, resulting in the display of "minimum safe" altitudes, which changes the nature of what needs to be comprehended (the pilot only must know his or her position with respect to that altitude, a one-dimensional, relative task).

INTRODUCING AUTOMATION TO THE FLIGHT DECK

In addition to research concerning information display and assessment, much was being learned about automation in general. The Fitts report (1951b) mentioned the need to understand how tasks should be allocated between human and machine, and this was the subject of some effort on the part of Fitts himself. Computer technology was in its infancy, and little was known in the 1950s about the possible limits on the ability of machines to assist humans. Many thought that these machines would gradually replace all human functions. Today, researchers better understand the different ways in which we can describe the relationship between human and operator, although this relationship is changing as new types of automated assistance are being added to aircraft and air-traffic systems.

This section will discuss the concept of allocating roles to automation and humans, beginning with some of the early concepts, and then progressing to more recent characterizations of the roles of humans and automation. Following this will be a discussion of how cognitive engineering has helped shape early systems, consisting mainly of control automation, and how it is significantly involved in newer types of systems, consisting mainly of information automation.

Autopilots

In a sense, automatic control capability, such as an autopilot, neatly satisfied the prominent conception of the proper roles of

automation and human operators. The automation took over the routine work of simultaneously maintaining course, altitude, and speed. Humans had a new system to deal with, but could leverage their long-term memory capability, reasoning, and judgment. However, Fitts (1951b) also noted that humans are poor monitors, and even suggested a specific proper role of automation may be to monitor humans. The addition of a highly reliable system to monitor for long periods of time violates that principle.

The autopilots, in fact, were a source of a number of accidents. A much-discussed accident in which an L-1011 crashed into the Everglades because the autopilot was inadvertently disconnected is an example of the "graceful" degradation of the autopilot system. The automation fails with little or no notice to the flight crew, who are not good at monitoring it to start with. Several other incidents demonstrated that pilots can overrely on automation to perform its tasks (Parasuraman & Riley, 1997).

The autopilot's development began in 1914, when Lawrence Sperry created a gyroscopic device capable of stabilizing an aircraft's flight without input from the pilot. This lead to the design of a fully automatic aircraft by 1915, and a fully automatic pilot by 1916, although it would not be until 1931 that an autopilot was licensed to operate with passengers on board (Davenport, 1978). An automatic landing system was invented in 1937 by Capt. Carl Crane, and since, numerous refinements in control theory have been applied to autopilots. One particularly interesting addition to control theory was the finding that the behavior of human-machine systems with simple feedback control systems (such as an aircraft under nominal flight conditions) can be modeled using a simple control law (McRuer & Graham, 1965).

While changes to the underlying control theory remain transparent to pilots, changes in the capabilities of the autopilots have not. Pilots interfacing with the autopilot often face "mode confusion," a term that indicates when the pilot's understanding of the mode in which the autopilot is operating differs from its actual mode. When this occurs, incidents such as those described above are more prevalent.

In general, however, autopilots have been amazingly successful. They relieve pilots of a great deal of manual workload, particularly during routine phases of flight such as cruise. Autopilots are also capable of increased accuracy at navigation, which is reflected in their ability to land aircraft in poor weather, including when poor visibility completely obstructs the pilot from seeing the runway.

Role Allocation

The "Fitts list" shown in Table 36.1 served as the launching point for studies on the allocation of tasks to humans and automation.

TABLE 36.1. Fitts List

Humans Are Good At	Machines Are Good At
Sensory functions, detection	Speed and power
Perceptual ability	Routine, repetitive work
Flexibility, improvisation	Computation
Judgment and selective recall	Short-term storage
Reasoning	Simultaneous operations
Long-term memory	Short-term memory

Automation for the control of aircraft had been around for decades by 1951. Not much more was automated, however, until the late 1960s and early 1970s, when some systems, navigation, and automated alerting were added. The early 1980s saw the advent of graphical displays, flight management systems, and the removal of flight engineers. The late 1980s introduced the concept of glass cockpits, replacing nearly all electromechanical "steam gauges" with graphical displays (Billings, 1997).

Over the decades, several different attempts at methodologies for allocating function have been made. This section will be organized around those efforts: (a) the Fitts list, (b) the "automate everything" philosophy, (c) supervisory control, (d) guidelines for automation, and (e) levels of automation.

Fitts list. In order to discuss the advantages and disadvantages of this approach, but without rehashing Fitts' discussion of them, we will first discuss the contents of the list, and findings surrounding the truth or fallacy of the claims. Then we will give examples of applying the Fitts list to automation, examining the strengths and weaknesses of the approach. Finally, I will give some examples of criticism levied at this method of function allocation.

Many of the items in the list still appear to be true. Very little argument can be made with the assertion that machines can generate more power than humans can, and they can perform many tasks faster than a human. On the other side of the list, humans are still much more capable of improvisation and making judgments, and computers are not able to reason in any meaningful way, despite significant advances in computer technology.

Yet some comparisons appear to be more complicated than suggested by the Fitts list. While a human's perceptual ability is quite impressive, some machines are capable of remarkable feats of detection: thermal imaging systems that can "see" through walls, satellites that can read individual license plates, microscopes that can image individual atoms, explosive detection equipment that can "smell" tiny amounts of specific chemicals, and so on. Despite this, it still requires a human to interpret the results. Machines still have only primitive (and brittle) abilities to deal with recognition (particularly of symbol systems, faces, and expressions, etc.).

In general, it appears humans are "wired" for rapid operation in a highly uncertain and diverse environment, whereas machines are still generally relegated to rather specific operations in a well-regulated and defined environment. Therefore, while machines may be better at routine and repetitive work, they are generally unable to deal with events or circumstances outside of the expected operating regime. Humans, however, are able to operate in the face of such events, leading to the notion that humans are suited for "supervising" automation (which will be discussed in a later section).

In practice, Fitts lists have been difficult to apply in allocating function (Sheridan, 1998). One likely reason is that a Fitts list approach requires automation to be assigned to a particular function. In reality, most functions require that some aspects are done by human operators and some by automation. Since the functions are shared between humans and automation, any distinction of human or machine for a function is likely to be suboptimal.

The "automate everything" philosophy. Although this method of allocation seems unjustifiable on its face, it has had its proponents, and its reasons. Humans have typically been seen as the source of a majority of error in the system. Replacing them with "trustworthy" automation, which also came with the benefits of lower cost and more efficient operation, did not seem quite so ridiculous several decades ago. Officials of two different aircraft manufacturers have been quoted as adhering to some form of this approach even within this last decade (Billings, 1997). In addition, these thoughts came at a time when the growth of technology appeared to be facilitating the design of automation.

By the late 1970s, it was becoming apparent that humans and automation had trouble getting along. Among the problems frequently cited (Wiener & Curry, 1980) were:

- Crew errors exacerbating automation errors
- Improper setup of automated systems
- Improper response to alerts
- Failure to monitor
- Loss of proficiency

In truth, however, the hurdles for an entirely automated aircraft have been, so far, too great to be surmounted. For one, the current social and political climate would have to undergo significant changes. Furthermore, since that time it has become apparent that automation is not as reliable as had been believed at the time. Examples of failures in automatic equipment have been common and persistent (Wiener & Curry, 1980).

In addition, under this strategy the operator would be left with any functions that could not be automated (Parasuraman, Sheridan, & Wickens, 2000). The result is that ill-defined, difficult tasks may be assigned to an operator that is significantly detached from the operation of the system. It is unlikely that the operator could perform well under these circumstances.

In such a system, the operator may be left with the task of monitoring the system with the resumption of control should the automation fail. Unfortunately, the skills necessary to do so would likely be degraded from nonuse. In addition, retrieval from long-term memory is more difficult if information goes unused for long periods of time, and the model of the system required for diagnosis and action is unlikely to be present in an operator who rarely actively controls it. These last few points have been referred to as "ironies of automation," since the more advanced the automation, the more valuable may be the operator's function, but the less able the operator may be to fulfill that function (Brainbridge, 1982).

Human supervisory control. About the same time as the automate-everything approach, a concept of interaction between humans and automation was being discussed, called human supervisory control (Ferrell & Sheridan, 1967). This was an offshoot of research into how humans could control a remote vehicle (specifically, vehicles on the moon). It became apparent that under the three-second delay for information to be sent to and return from the moon, some control loops would have to be closed by the automation, with the operator acting as a supervisor of the automation (Sheridan, 1992).

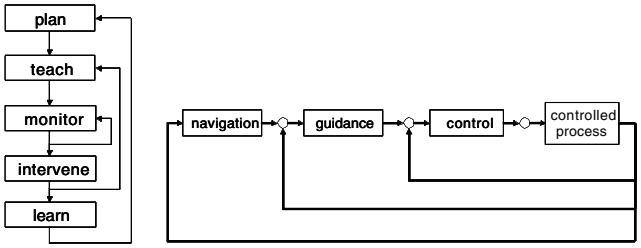


FIGURE 36.2. Sheridan's supervisory and control loops. (Adapted from Thomas B. Sheridan. *Telerobotics Automation, and Human Supervisory Control,* Figures 1.2 and 1.3, Massachusetts Institute of Technology, published by The MIT Press).

The concept reflected the growing role of automation, and the changing of the human's role from operator to manager of automation. As shown in Fig. 36.2, several levels of supervisory and control loops exist, with the human closing the supervisory loops, and each control loop being able to be closed by either the automation or the human.

Designers could now employ control theory concepts to the human-automation problem. The role of humans was essentially defined by this paradigm, and the designer could then look at lower level tasks and turn to questions such as (a) what human resources are required for each task, (b) how much time and effort is required, (c) how much time and energy is available, and (d) what the consequences are of accomplishing or not accomplishing an action (Sheridan, 1992).

Sheridan goes on to describe the limitations of humans and automation, and the implications of this for applying the concept. These admonishments went unheeded, however, and automation was steadily added to the flight deck and control rooms, with the expectation that the operators would handle the additional managerial workload.

Levels of automation. Parasuraman, Sheridan, and Wickens (2000) recently proposed a new model for allocation of automation functions across four classes and along a continuum of low automation to high automation (Parasuraman, Sheridan, & Wickens, 2000). The four classes are (a) information acquisition, (b) information analysis, (c) decision and action selection, and (d) action implementation. The level of automation can be allocated for each of these four levels, and can be adapted "on the fly" during operational use. The overriding consideration in determining level of automation is the consequences for human performance in the wake of that automation decision.

Information acquisition automation relates to the "sensing and registration of input data." This includes such things as moving sensors (low level of automation), organizing information (moderate), highlighting information (moderate), and filtering information (could be a high level of automation).

Information analysis automation relate to the requirement to integrate and/or transform information. Low-level examples include showing track predictions or trends. At higher levels, data may be integrated, displaying only high-level information to the operator.

Decision and action selection automation assists in determining courses of action and in making decisions. Low-level au-

tomation in this class may simply recommend courses of action. At higher levels the automation may perform the action that it feels is appropriate. The difference is demonstrated by the ground-proximity warning system (GPWS), which recommends an action and is at a fairly low level, and the ground collision avoidance system, which is at a significantly higher level as it automatically takes control to avoid a ground impact if the pilot does not take action.

Action implementation automation performs actions for the operator. The level is determined by the amount of manual control left to the human operator. For example, an autopilot on heading hold may still require heading changes using a turn controller and speed control using throttles. This would be a lower level of action implementation than a fully coupled autopilot.

The appropriate level of automation for different classes should then be decided upon by examining possible solutions at different levels of automation and applying primary and secondary criteria in a possibly iterative fashion. The primary criteria are based upon aspects of human performance such as mental workload, situation awareness, complacency, and skill degradation. The secondary criteria are based upon more practical concerns, such as reliability of the automation and the costs versus benefits of automation decisions.

Ecological allocation. Advocates of this approach (although they do not use this term to describe it) regard function-based representations of work as inadequate representations of the actual workplace. The workplace is complex, and there is interplay between technology and work—that is, just as the workplace is changed by automation, the use of the automation may be changed by the workplace. The claim is then that one cannot adequately allocate function without fully comprehending the work environment and the situations that arise therein. Suchman (1987) stressed that work is "situated," and as a result the system must "incorporate both a sensitivity to local circumstances and resources for the remedy of troubles in understanding that inevitably arise" (p. 28).

Moreover, the situations that arise can fundamentally change the work environment. In a discussion of previous work, Wright, Dearden, and Fields (1998) related three themes that emerged from their (and other related) studies:

> The first is that divisions of labor are often dynamically re-allocated on the basis of local contingencies on an occasion of practice. The second is that the processes by which such working divisions of labor are achieved are a significant component of that work. The third theme is that even where divisions of labor are pre-computed and reified in such mechanisms as standard operating procedures, there is still work to be done in making those procedures work. (Wright et al., 1998, p. 339)

Often, due to the changing nature of work, organization, and even situation, the underlying functions of automation can change. In a study of two incidents in the London Underground (subway system) in which trains left their drivers behind, it was discovered that the operators had altered one of the two controls required to operate the train (Wright et al., 1998). One automated control requires all doors of the train to be shut before the train will move. A second control is simply a button for the operator to push to start the train moving. The operators, apparently believing that the second control was redundant, taped the button in the on position. The two incidents happened when operators left their station to fix stuck doors. When the doors were closed, the "circuit" was closed, and the trains began moving.

Wright et. al. (1998) argued that the operators had unknowingly changed the allocation of functions. Their function had been removed—the automation now had sole control to operate the vehicle. Yet these types of changes to function are not only the result of errors. Pilots make use of automation in ways that change their function as well (e.g., stall warning systems as angle of attack indicators, situational awareness instruments as navigation sensors).

For this reason, it may be desirable to allocate functions dynamically, as it is done in normal social work contexts. In this concept, the allocation of function would depend on the context of each occasion of use (Scallen, Hancock, & Dudley, 1996). In this case, designers would have to examine the consequences of improper configuration of the automation and how to mitigate the likelihood of that occurring (Dearden, Harrison, & Wright, 1998).

One method of obtaining the richer representations of the work context called for in this concept would be the use of scenarios (Dearden et al., 1998), as discussed in chapter 54. Other methods include ethnographic approaches (Hughes, Randall, & Shapiro, 1992) and contextual design methods (Beyer & Holtzblatt, 1998), which are discussed in detail in chapters 51 and 50, respectively. The former finds researchers immersing themselves in the work environment under study to attempt to understand the social organization of the work. In the latter, the work environment is viewed from different perspectives (e.g., user, cultural relationships, sequences of actions, information artifacts, physical layout) to define roles and tasks.

Guidelines for automation. Over the years, a number of efforts at establishing guidelines for automation have been described. Several of these, by notable authors, are discussed here.

Wiener and Curry first presented guidelines for automation in 1980. They advised designers to create automation that was comprehensible to the operator, so that the operator could detect and properly react to failures in the system. They also advised that the system should appear to perform the task in much the same way as the operator, allowing for individual styles of operation while not affecting overall system performance. They suggested that operators should be well trained on the setup of the system, as well as nominal and off-nominal operation, emphasizing its use as an aid rather than as a replacement. Operators should also be trained on the alerts of the systems, which should be able to be validated by the operator, and should clearly indicate both the source of the alarm and the severity of the situation. Wiener and Curry also suggested that monitoring may be an important task for the operator, who should be trained and motivated to monitor, and whose workload should be kept at a moderate level to avoid both high- and low-workload situations.

Billings, in 1997, agreed with many of these findings, but stressed that the automation should never have control over the system. The operator should be in control, and even should not

be automatically prevented from exceeding normal operating limitations. Billings also warned against overautomation, the tendency of complex automation to become incomprehensible, and the resulting system being too complex for the operator to control in the case of automation failure.

Parasuraman and Riley (1997) added to these guidelines by pointing out the possibility that in certain situations operators can "over-rely" on automation, while in other situations operators can "under-rely" on the automation (Parasuraman et al., 2000). These situations generally occur when the operator does not have sufficient knowledge about the automation or the situation, or when confidence in the automation is misplaced (either high or low). They found that sudden alerts do not adequately allow the operators to prepare to respond, and suggested the use of preparatory, or "likelihood," alerts. They warned that in many cases, automation replaced operator error with designer error, which can occur if designers do not incorporate knowledge of operational practice into their design, and do not allow for some nonintended uses of the automation.

Early Air-Traffic Control

Efforts at controlling air traffic began quickly after the advent of aircraft. These early efforts consisted of drafting procedures for safe operation of aircraft, and were undertaken by private companies until the 1930s (Gilbert, 1973). As aviation grew, dedicated professionals increased their efforts to track aircraft and ensure their separation from one another.

This process was, and to a surprising degree still is, a highly manual process. Initially, aircraft had contact via radio with their companies, who relayed position and timing information to a central control facility. The controllers in that facility would record this information on a large blackboard and manipulate aircraft markers (also called "shrimp boats") on a map table. They would analyze this information in an attempt to predict potential conflicts between aircraft. This ability to mentally project aircraft states is still relied on today when radar displays fail.

Progress from these modest beginnings was slow compared to the progress of aircraft technology. The two most significant advances in air-traffic technology occurred after World War II. At least partially due to their experiences in the Berlin Airlift, air-traffic personnel adopted the practice of speaking directly to aircraft. In addition, the development of radar allowed controllers to track aircraft in real time.

Early radar displays. Although the application to air-traffic control is obvious, early efforts at controlling aircraft with radar were troubled. Early radar displays showed the position of each aircraft relative to the radar station, but did not display other critical information, such as aircraft identification, speed, or altitude. Because of this, associating targets with radar returns was difficult, and projecting their status into the future was still a largely cognitive process. The recognition of this helped bring about the introduction of secondary surveillance radar, in which a transponder on board each aircraft transmits an identification code and altitude to a ground receiving station (Gilbert, 1973). This information can then be correlated with the radar returns, eliminating the need for map tables.

Flight strips and strip bays. Controllers gradually replaced the blackboard with small strips of paper on which information about each flight was recorded. These would be placed in a rack (or bay) in temporal order, and even passed from controller to controller as the flight progressed. As computers were added in the 1960s and 1970s, one of their first tasks was to automate this flight data processing, including printing flight strips.

MODERN CHALLENGES, SOLUTIONS, AND PRINCIPLES

The first 50 years of aviation saw significant changes and innovations in both aircraft and air-traffic control. Recent developments such as (a) complex flight management systems, (b) multifunction displays, (c) highly integrated information systems, (d) alerting and warning systems, and (e) sophisticated automatic control capability have presented even more complex challenges for aviation professionals and researchers.

Glass Cockpits, Flight Management Systems, Mental Models, and Mode Confusion

Beginning in the 1970s, airliners were equipped with flight management computers, which would assist pilots in numerous tasks, including flight path management, navigation, and fuel burn management. The interface to the flight management computer is through a control display unit, which consists of (a) a small, alphanumeric keyboard, (b) several function keys, and (c) a small display screen, all of which fits into about a 6-inch by 10-inch space. On this unit, pilots must enter navigation information, select information to be displayed, and navigate through menus. By the 1980s, flight-management computers were integrated with cathode ray tube (CRT) displays, creating flight decks mostly devoid of gauges, also called "glass cockpits." An example of a glass cockpit (from an MD-11 aircraft) is shown in Fig. 36.3.

Such systems integrated the operation of the aircraft with navigation, resulting in a "fundamental shift in aircraft automation" (Billings, 1997). Although such displays provided the benefits of a highly integrated display, there was a cost in terms of workload, crew coordination, error management, and vigilance (Wickens, Fadden, Merwin, & Verves, 1998). In addition, a number of accidents have been associated with pilots misunderstanding the state of the automation. These accidents have been attributed to the pilots lacking an accurate "mental model" of the automation.

Mental models, described in detail in chapter 4, are a conceptual representation of a system posited cognitively. The precursors to mental model research date back nearly a century. These early inquiries (Wittgenstein, 1922; Craik, 1943), and a significant body of recent work, argue that cognition is inherently imagistic, so that mental models are a theory of how people think. This theory is a topic of contentious debate in the psychological community, with many researchers vehemently denying that people think in these "pictures in the head," but rather that thought consists of formal rules of inference (Pylyshyn, 2003).

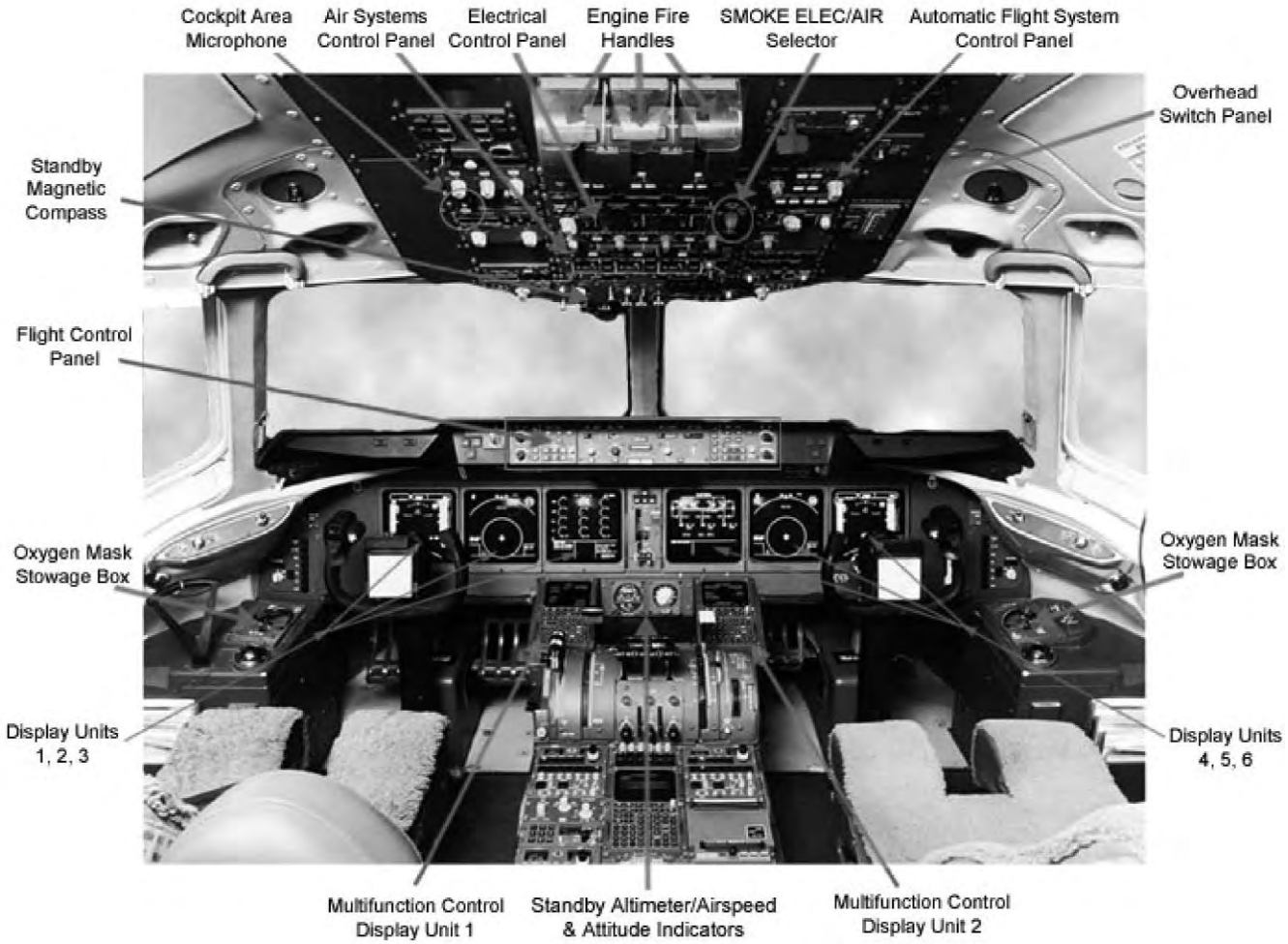


FIGURE 36.3. MD-11 flight deck instruments. (Photo from Transportation Safety Board of Canada Report Number A98H0003, page 32, Transportation Safety Board of Canada, 1998. Reproduced with the permission of the Minister of Public Works and Government Services, 2007).

Undeniably, however, people at least think about such imagistic models. In doing so, operators reason about their system based on these models. There is convincing evidence that such models correlate well with certain operator errors, and point to ways in which good design can mitigate these errors. As such, we will try to skirt the contentious issue of whether mental models are theories or models, and stick to how the concept has been deployed in support of cognitive engineering.

Mental model research was borne out of the observation that many people use analogy to understand complex phenomena. This idea became more formalized in the cognitive science literature, highlighted by a flood of papers and books on mental models in the early 1980s (Gentner & Stevens, 1983; Johnson-Laird, 1983). If one believed that a person's knowledge of a complex system was a product of that person's model of the system, then an understanding of the limitations of these models was crucial to explaining behavior and errors.

Nowhere did mental models seem to fit in better than in explaining a number of human-automation interaction failures. Flight deck automation was becoming increasingly complex, taxing the abilities of pilots to accurately model it. Error is likely where such inaccuracies exist. Several particularly prominent examples of this have occurred in the last 15 years. In 1988, a Boeing 767 was inadvertently put into a vertical speed mode, nearly resulting in a crash. A similar situation occurred in an Airbus 320, when a crew mistakenly selected a descent rate of 3,300 feet per minute instead of on a 3.3° downward slope and flew into terrain. In 1994, a China Airlines Airbus 300 aircraft crashed after pilots tried to continue a landing while the autopilot had been inadvertently put into a go-around mode. Also in 1988, an Airbus 320 was flown into terrain by an aircrew during an air show; the crew believed that the envelope protection of the aircraft would prevent a stall, but the aircraft was in a mode where that protection was not provided.

These incidents indicated that one piece of flight deck automation particularly prone to such mental model inaccuracies is the flight management system, which interacts closely with the autopilot. Researchers had been aware of mode problems in HCI for some time, and in due time these issues arose in the interaction of the pilot with the computer that operates the

autopilot. This aspect of the problem—as a computer interface issue—has significantly affected the research on mental models in aviation.

In addition to calling for training for pilots to better understand their systems, much effort has been put into creating models of the pilot's interaction with the flight management system in an effort to understand how systems might be designed to mitigate these model inaccuracies.

Heads-Up Displays

In order to minimize the need to view instruments inside the cockpit, "heads-up" displays (HUDs) were superimposed on the windscreen, providing at least the basic T information. The same method can be used to superimpose images on the inside of the visor of a helmet (known as "helmet mounted displays" [HMDs]). HUDs have moved from the military, where they found their first application, to commercial aircraft and even automobiles.

One major difference between the instrumentation on a HUD and the same instrumentation on a traditional instrument panel is that the world can be viewed through the HUD. This allows for the possibility of integrating real external images with the instrumentation without violating the proximity principle. The flip side of this, however, is that the world peeking through the HUD provides a great deal of clutter on the display, although some research indicates that this is not a significant increase in difficulty (Ververs & Wickens, 1998). Overall, the HUD has a number of advantages that make it superior to conventional positioning of displays. The HUD enhances the operator's ability to switch between near and far domains (Levy, Foyle, & McCann, 1998), although strangely this reverses to a detriment for the detection of unexpected events (Wickens & Prevett, 1995).

Information Automation

The rapid rise in computer and display technology has allowed engineers to develop systems that alleviate some of the information burden on the operator. These systems perform automated monitoring (such as conflict alerting and prevention), provide information to assist the pilot's decision making, and help provide coordination between different operators.

Alerting systems. In the late 1970s, aircraft were undergoing a rapid expansion in terms of warnings and alerting systems. One survey found general agreement that there were too many warnings and that standards and guidelines were needed for warning system design (Cooper, 1977). Another researcher determined that the number of visual alerts and flags had doubled, while the number of aural alerts (see chapter 14) had increased by 50% since the beginning of the commercial jet age (Veitengruber, 1977). Based on the data at the time, researchers were recommending that alerts be reduced, prioritized, and inhibited during critical phases of flight.

Since then, alerting systems have grown significantly in capability and complexity. Current alerting systems may utilize multiple sensors, ground communications, large knowledge databases, and may have sophisticated algorithms for alert threshold and guidance recommendations (Pritchett, 2001).

One of the most widely analyzed problems in alerting system technology relates to problems with conformance to the ground-proximity warning system. The ground-proximity warning system (GPWS) was a complex alert designed to warn of something extremely hazardous: proximity to or closure to terrain. It was complex in that closure to terrain can be caused by a number of things, including landing, and the system had to discriminate these events. In part due to this complexity, and in part due to poor design, the system frequently produced false alarms. This had the effect of reducing operator trust in the automation and causing underreliance (the pilots would not believe the alert).

High false alarm rates can also be caused by improper setting of thresholds for alerts, a process that is analogous to signal detection. The threshold for an alert can be based upon the system operating characteristics curve, which is in turn a function of the sensors (Kuchar, 1996). Setting the threshold is then a tradeoff between missed detections and false alarms. Typically, this means that the fewer the false alarms, the more the missed detections. Interestingly, some researchers have warned against having too low a false alarm rate (Farber & Paley, 1993). If the false alarm rate is too low, not only will the occurrence be infrequent and unexpected, but also it will likely require a later alert, compounding the difficulty of reacting to the alert.

The problem of false alarms is further exacerbated by the low base rate of incidents requiring an alert (Parasuraman & Riley, 1997). In these cases, the relative rate of false alarms is increased. Yet in many cases, the cost of a false alarm is considered much lower than a missed detection, leading designers to err on the side of economy and/or safety.

These (generally) are cases where the alerting system is functioning as a signal detector or a hazard detector, the latter being a more complex version of the former. Alerting systems have also added hazard-resolving logic, such as in TCAS II (Pritchett, 2001) and NASA's Advanced Airspace Concept (Erzberger, 2004). The benefit of such an approach is that the resolving maneuver can be incorporated into the alerting logic, resulting in a reduction of false alarms; however, such a system relies heavily on the pilot appropriately performing the maneuver. It is also possible that the maneuver may actually induce a collision by creating a problem where none actually existed.

Decision support systems. In addition to alerting systems, there are also systems on board aircraft and, to an even greater extent, in air-traffic control which are considered decision support tools (see chapter 30 for a discussion of design approaches for decision support systems). These tools do not warn or alert, but provide information to support decision making. This information is often integrated from several sources, or enhanced in some way.

One simple example of a decision support system on a flight deck is the weather radar system. This system provides no discrete alert, but rather shows the pilot where hazardous weather exists to help pilots determine how to deviate to avoid it. With no weather radar, pilots had to rely on visual means, which can only detect gross features such as height, darkness, and the presence of lightning. Weather radar, however, not only shows

rates of precipitation (an indication of severity), but can also show the lateral and vertical extent of the weather.

Another decision support system is the Engine Indication and Crew Alerting System. While some aspects of this system are more correctly considered an alerting system, much of the system is simply decision support. This display shows pilots the status of most of the systems on board. It can be configured to depict engine instruments or other graphical depictions of the operation of the systems on board the aircraft.

As mentioned, there are a number of decision-support tools being used in air-traffic control. In fact, the number of such displays is increasing at such a rate that the FAA is attempting to integrate these systems to reduce the sheer number of separate systems operating in air-traffic facilities.

Modern Air-Traffic Control

The current air-traffic control system, while reliant on computer resources, has been designed to be resilient to the loss of such resources. Computers are still used for display of primary and secondary radar information, as well as flight data processing. The upgrading of these capabilities from the 1970s has been slow, impacted by the widespread replacement of experienced controllers following the air-traffic controller strike of 1981 and the failure of the Advanced Automation System (AAS) program of the late 1980s (Lee & Davis, 1996). On the heels of the AAS system, however, have come a number of advances, including attempts to automate flight strips and the introduction of automation aids.

Automated flight strips and situation awareness. Flight strips have proven remarkably resilient to change. Many controllers still use flight strips, and flight strip bays are still used in air-traffic control facilities. Recently, however, manual flight strips are being replaced with electronic flight strips, most notably in the User Request and Evaluation Tool (URET), which took 25 years to implement (Arthur & McLaughlin, 1998). Similar efforts to replace flight strips have been undertaken in Europe (Berndtsson & Normark, 1999). The reasons for their slow demise are highly instructive of some of the problems with automating some forms of human work.

Despite the significant workload of the manual process of dealing with flight strips, controllers have been reluctant to part with them, or to use suggested replacements. This is, in large part, due to the flight strips contribution to the controller's situation awareness. For a controller, whose job is still to a large degree cognitive, their situation awareness is crucial (Whitfield & Jackson, 1983). Researchers have found that controllers indicate that flight strips are a primary means of establishing and maintaining their situation awareness (Harper & Hughes, 1991).

It is a notable theme that modern research attention has turned from information display, as described above, to information assessment. As more complex automation is added to flight decks and air-traffic facilities, and as more information is provided to operators, accessing and comprehending that automation and information became problematic. Researchers began to understand that not only did they need to know how to display information, but also how operators represented and utilized that information, in order to provide a better match between technology and the human.

Interest in information assessment has lead to a significant body of research into situation awareness (SA). This has been a particularly important concept in aviation, since safe and successful operation in aviation requires a great deal of knowledge about the environment outside of an individual aircraft. Many aviation accidents can be attributed to a lack of knowledge of the environment external to the aircraft (e.g., collisions with aircraft and the ground). Air-traffic controllers' knowledge of the "big picture" includes not only the states of the aircraft in their own sector, but those of neighboring sectors, airports, weather, and much more. Attempts to understand how pilots and controllers obtain and maintain this "big picture" have been the focus of SA research.

Since as far back as the first World War, military aviators have understood the importance of SA and have instructed their pilots in developing and maintaining good SA. One of Germany's top World War I fighter aces, Oswald Boelcke, listed among his "dicta" that "the pilot must acquire the habit of "taking in" unconsciously the general progress of the whole multi-aircraft dogfight going on around the individual combat in which the pilot will become involved . . . (so that) no time (is) wasted in assessment of the general situation after the end of an individual combat" (Hacker, 1984, p. 1). Boelcke also prescribed knowledge of one's own machine, the enemy's machine, and navigational fixes. SA has been a significant part of aeronautical training since that time.

As a research topic, however, the concept was mostly ignored by researchers until the 1980s. Due to increasing flight deck and air-traffic automation, pilots' and air-traffic controllers' role as supervisor of these systems was increasing, reducing the time they could spend in developing SA. At the same time, a great deal of new sensor information was becoming available to designers of aviation automation, information that could be used to reinforce the controllers' and pilots' SA. Researchers began looking into what SA is, what affects an operator's ability to construct SA and to keep it, and how it might be measured.

Many definitions of SA exist, with most agreeing that SA is, at least in part, the comprehension of elements of the environment that have (or may have) some bearing on the task being accomplished. Some efforts have viewed SA as a static, information-driven product, some have viewed it as a dynamic process, while others have viewed SA as a high-level description of certain aspects of task behavior. A significant body of work has also gone towards determining how to measure SA.

One of the most widely quoted definitions of SA is "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley, 1988, p. 97). Another researcher has called SA "an integrated understanding of factors that will contribute to the safe flying of the aircraft under normal or nonnormal conditions" (Regal, 1988, p. 65). These definitions describe SA as a product, something that an operator either has or does not have. As such, errors resulting from a lack of SA can be studied. Researchers have found that controllers, viewing recorded air-traffic files, were unable to recall many details about traffic when asked about them during

the scenario, including call signs, control level (who has control of the aircraft), altitude, speed, and heading; controllers often even failed to report that some aircraft were present at all (Endsley & Rogers, 1996). Other researchers studying error report databases have found that in only about 20% of serious operational air-traffic errors were controllers aware of the problem developing (Gosling, 2002).

Researchers who view SA as a product often defer the process of obtaining and maintaining SA to a separate process, often referring to it as "situation assessment". This type of SA has been described as "(an) adaptive, externally-directed consciousness" (Smith & Hancock, 1995, p. 137), and "the integration of knowledge resulting from recurrent situation assessments" (Sarter & Woods, 1991, p. 45). Here, the sources of information important to SA (visual, auditory, tactile, other sensory input, knowledge of procedures and regulations, etc.) are diverse and often clearly identifiable. The operator's ability to obtain SA begins with the most fundamental cognitive processes (detection), and progresses to very sophisticated concepts (comprehension and projection).

Some researchers have called into question the utility of the SA concept. Many feel it is a high-level concept that does not have sufficient granularity to really explain anything. SA has been referred to as a description of observations of humans operating complex systems in a dynamic environment (Billings, 1995) and as simply equivalent to expertise (Crane, 1992). As such, SA is a description of a set of cognitive processes that are used together, and is really only useful in categorizing or grouping behaviors and errors. Some researchers feel that considering SA as a causal agent can be counterproductive to understanding operator behavior (Flach, 1995).

No one set of measures has been clearly identified for SA. Typical measures can be broken into three categories: (a) explicit (or knowledge-based), (b) implicit (or performance-based), and (c) subjective (Vidulich, 1992). Examples of explicit methods include participants' recall of a situation (post hoc), an ongoing narrative provided concurrently with the task, or freezing a scenario and questioning the participant about the decisions, events, or the task environment. These measures can be compared to actual state of the system to provide a better measure, but have been criticized as too subjective (in the case of recall; Fracker, 1991), or too intrusive (in the case of freezing a scenario; Sarter & Woods, 1991).

Implicit measures examine task performance, and correlate that with SA. These measures are generally unobtrusive and objective, and (as mentioned earlier) can be used in conjunction with explicit measures. It has been suggested that these measures can succeed where explicit methods cannot, particularly in situations where a determination of the timing of events is important and where the subject may be unaware of his or her deficiency (Pritchett, Hansman, & Johnson, 1995). These measures may examine performance at the task overall, or alter the task to determine whether the subject notices the change.

Subjective ratings are assessments of SA made either by an observer or by the subject. These ratings can use a number of different scales, and can be either direct or relative (such as by comparing SA in one situation to SA in another situation). Although these ratings can be affected by a number of factors, including task performance, one technique that has been studied extensively is the Situation Awareness Rating Technique (SART; Taylor, 1989. This technique has the operator rate 10 constructs on a seven-point scale. The constructs are grouped into three categories: (a) attentional demand (which includes the instability, variability, and complexity of the situation), (b) attentional supply (including arousal, spare mental capacity, concentration, and division of attention), and (c) understanding (which includes information quality and quantity, and familiarity). SART has been found to be more sensitive than overall subjective measures of SA (Selcon & Taylor, 1989), although some researchers feel that SART confounds SA with workload (Jones & Endsley, 2000).

Decision support tools in air-traffic control. One of the most significant areas of improvement within air traffic over the last two decades has been in decision support tools. Over the years, a number of automation tools have been slowly added to the air-traffic control system, and others have been considered and failed. These advancements have included the introduction of two conflict prediction systems, the failed attempt to provide sequencing and runway selection aids, and the introduction of centrally scheduled time-based metering tools.

The first conflict prediction system was created in the early 1980s as a quality assurance program. The "operational error detection program" (OEDP) was designed to identify when two aircraft had come closer to one another than allowed under FAA regulations. If the OEDP was set off, a record was made of the transgression, and investigation took place. This is obviously highly undesirable for controllers, who have nicknamed the system "the snitch patch." As a result of the OEDP, controllers routinely pad separation between aircraft to ensure that the snitch patch does not activate (Cotton, 2003).

The OEDP also gives a three-minute warning that a violation is going to occur, although the system is not considered highly reliable by controllers. Part of the reason for the system's unreliability is that its predictions are based on simple extrapolations of the current states of the aircraft involved, without consideration of intentions (such as its intention to stop climbing or to change course). The system's unreliability has led to several efforts to replace it. The URET system mentioned previously includes a "conflict probe," which examines the predicted flight path of aircraft and gives an indication of the potential for collision (Brudnicki & McFarland, 1997).

In the 1990s, NASA undertook an effort to provide automation tools for controllers and traffic managers. The suite of tools, called the Center-TRACON Automation System (CTAS), was based upon combining radar position and speed data, aircraft flight plans, weather data, and models of aircraft flight characteristics. The models were based upon those used in the flight management systems in aircraft, which are known to be highly accurate. One of the first tools proposed was the Final Approach Spacing Tool (FAST). This system was to be used by controllers handling aircraft close to busy airports. These controllers transition aircraft from intermediate altitudes and about fifty miles out from an airport and coordinate their arrival to the runway. This is a very complex task, requiring controllers to predict the proper sequencing and merging of aircraft from several directions. FAST generated advisories for sequencing, runway assignment, headings, and speeds to ease the merging and spacing of aircraft (Lee

& Davis, 1996). Although deemed acceptable by controllers (Lee & Sanford, 1998), the system was ultimately not implemented. One of the likely factors was that although the system was extremely accurate in its predictions, it was not perfectly so, and controllers may have felt that the workload and distraction of the few inaccurate predictions outweighed its benefits.

Another part of the CTAS suite of tools is the Traffic Management Advisor (TMA), which provides a similar service as FAST for controllers handling aircraft further out from the airport (out to about 300 miles). This system provides delay advisories for each aircraft approaching a busy airport. Controllers are then responsible for slowing aircraft to meet the assigned delays, utilizing the same techniques they use to space aircraft normally. The main shift between the current method and TMA is that the spacing TMA provides is time-based, versus the distance-based method currently used. Time-based metering in this fashion has been shown to be more efficient than distance-based (Sokkappa, 1989; Vandevenne & Andrews, 1993), and TMA has shown significant benefits where it has been adopted (Knorr, 2003). Similar efforts focused on runway sequencing and scheduling have been developed in Europe, Canada, and Australia (Ljungberg & Lucas, 1992; Robinson III, Davis, & Isaacson, 1997; Barco Othogon, 2003; NavCanada, 2003). More advanced systems are also being developed which move some separation authority to the flight deck (Prevot et al., 2003).

FUTURE CHALLENGES

Aviation has undergone a great deal of change since its inception, although predictions of its development path have not always been accurate. Forty or 50 years ago, predictions of where the air transport system would be would probably have included much more automation of air-traffic control and high reliance on satellite systems, with larger and faster aircraft than were then utilized. These aspects have largely not been realized to the extent expected. The aviation system of the future will undoubtedly hold other unexpected developments, but a few challenges for researchers, engineers, and practitioners are clear.

Three-Dimensional Displays

To overcome the limitations of two-dimensional (2D) displays, several alternatives to providing three dimensions in a display have been investigated. Air-traffic and navigation displays are both trying to represent three dimensions in a 2D display. Inevitably, some information is lost in the display in this case. This can be accomplished through binocular, stereoscopic, or holographic devices (Mountford & Somberg, 1981), but practical problems make this unlikely. Also unlikely due to the cognitive demands of integration is using time frame compressions (a display of sequential images; Roscoe, 1981); however, significant research has been accomplished in using perspective illusion displays, where the illusion of three dimensions is given through the use of perspective.

Three-dimensional displays can portray a variety of viewpoints, to different effect (Wickens & Prevett, 1995). In testing such displays, 2D displays, with either some analog display of altitude or an additional profile view, are used for comparison. This adds complexity to testing these displays; often, additional information has to be added to current displays in order to isolate the 3D presentation from the additional information provided by the display.

In addition, dual panel 2D displays, although they may contain the same information as 3D displays, have an additional cost—that of scanning and integrating. Therefore, it would seem that 3D displays adhere to several tenets of good display design: proximity compatibility, integration, and pictorial realism.

If the airplane is not navigating vertically, however, the 3D display may impose a clutter cost, as the operator must sort through the irrelevant altitude information. This is in addition to the problem of ambiguity of position of objects along the line of sight of the display (McGreevy & Ellis, 1986). Lastly, if the 3D perspective is immersed (the perspective is as the world looks from the cockpit), no information concerning the world to the side or behind the aircraft is presented, resulting in a "keyhole" view of the world (Woods, 1984).

For judgments requiring mapping between the world and the display, a 3D display with a viewpoint outside the cockpit seems to provide the best performance (Olmos, Wickens, & Chudy, 1997). This benefit is again tied to the principles of realism and integration, and is generally viewed as providing better overall situation awareness than immersed 3D views.

Tunnels in the sky. Enhancements to the immersed viewpoint of a 3D display can present the operator with a highly integrated pathway, or "tunnel in the sky." Superior performance in navigation and tracking has been demonstrated using this display (Wickens & Prevett, 1995), although they are currently still in development. The biggest obstacle to the deployment of such automation seems to be the cost of certification, rather than any technical or cognitive issues. It appears that these displays only suffer from effects similar to those found on other perspective displays (confounding of distance along the line of sight) and on HUDs (difficulty in detecting unusual events due to the compelling nature of the display; Wickens, Mavor, Parasuraman, & McGee, 1998).

The scale problem in air-traffic control. Increases in computational ability have made 3D displays even more practical, but many issues surrounding them have not been resolved. In some ways, aviation is a 3D task, with vertical and horizontal navigation; however, the scales of these navigation tasks are different, one expressed in feet, the other in nautical miles. It is extremely difficult to portray these two axes on a single display without distorting one or the other, with significant repercussions for judgments and decisions of the operator using the display. In addition, air-traffic control is only marginally a 3D task, with aircraft separated for much of its flight through the use of altitude stratification. This separation method places aircraft at discrete altitudes, with climbing and descending transitions between these strata. As such, separation in the vertical axis can be assured for much of the aircraft simply by comparing two numbers. Additional information about vertical position is superfluous (except during climbs and descents).

Coordination Support Systems

One type of information system which is likely to become prevalent in the future is what is referred to here as a coordination support system (for a further discussion of computer supported cooperative work, see chapter 28). It is becoming more common that work, including in the air-traffic system, is distributed across organizations. For example, alerts generated by an aircraft's TCAS system have to be coordinated between the two aircraft in the conflict pair to avoid having the aircraft's resolutions conflict. In addition, coordinating arrivals into airports involves affecting aircraft many hundreds of miles away from the airport, across air-traffic control organizational and even facility boundaries.

One approach to enabling such cooperative work is to actively share information between organizations through procedural means. Such an approach means that one organization, with localized information of interest, distributes that information to other parties. In the air-traffic system, however, the great majority of the information needed for collaborative work is present in the task environment, or relates to the impact of other organizations' decisions on the traffic situation. If such ecological information could be provided, then collaborative work would happen naturally.

Such an approach is supported by the concept of a common information space (CIS) (Schmidt & Bannon, 1992). A CIS is a source of information shared within and across organization boundaries in order to conduct coordinated work for common purposes. Instead of passing information from one organization to another, information is made available through the CIS, which can then be accessed and manipulated by any interested party. Moreover, objects can be created within the CIS which serve as translators across different communities. These objects, referred to as "boundary objects," are entities in which decontextualized information resides in a particular form (Starr & Griesemer, 1989). The distinction which defines boundary objects is their purpose in acting as translators, or, as defined by Bannon (2000), they are a "means for sharing items in a common information space" (p. 5). Information, devoid of context, is presented in a form that allows users from different communities of interest to comprehend it within their own contexts. This object is accessed by users in different communities of interest in order to conduct their particular portion of the work. The information in boundary objects is contextualized only within each community of interest, and this context differs in each separate community.

Technologies Needed for 3× Air Traffic

The rate of growth of air traffic has been consistent for a number of decades, and is not expected to change significantly in the next few decades. Such a rate of growth would double or triple air traffic, a level which some researchers feel is unsupportable by the current method of air-traffic control.

Such a conclusion is consistent with the historical development of the air traffic system. In order to grow beyond its capabilities in the 1940s, radar was added, reducing the workload of the controllers, which enabled them to deal with more traffic and higher aircraft speeds. To continue to grow, computerized flight data processing was added, further reducing workload and allowing controllers to handle more traffic. Increases in staffing, reduction in sector sizes, strategic control, and other innovations have allowed further increases in air traffic levels. To further continue these increases, researchers argue that a more fundamental shift is needed, by sharing some of the air traffic responsibilities between the current authority (air traffic controllers) and the flight deck, by allowing automation to accomplish some current tasks of the controllers, or some combination of these.

Datalink and the loss of the party line. One high-workload task for controllers is communication. For some air traffic control tasks, the capacity of the radio frequency is a limiting factor for how many aircraft can be handled within a sector. This is also a limiting factor for how small sectors can be, since controllers must still talk to each aircraft at least twice (once upon entering, once upon leaving the sector), and small sectors reduce the time between these communications.

One solution to this communication workload is to allow communication directly between the data processing system and the flight, through a system known as "datalink." This technology has been suggested for decades (Schmidt & Saint, 1969), but its use has been stifled by several issues.

One of these issues is that pilots' situation awareness is affected by the availability of information on the communication channel. Pilots hear the communication of other aircraft with the controller, and can glean information that is relevant to them, such as expectations of future clearances, the impact of weather on aircraft ahead of them, and positions and intentions of other aircraft. If this "party line" information is passed over a data channel, pilots would no longer have access to it, reducing their situation awareness (Midkiff & Hansman, 1993).

4D contracts: Future changes in the roles of pilots and controllers? The flight management systems used by the current generation of commercial aircraft is capable of meeting arrival times over points in its flight path, subject to the constraints of the flight envelope. The logic behind these flight management systems is also available to ground-based systems, and is in fact the heart of the Center-Tracon Automation System, a suite of air-traffic management tools developed by NASA (Erzberger & Nodell, 1989).

These ground-based systems can use the flight management algorithms to make predictions of arrival times for all aircraft in the National Airspace System (NAS). This has been accomplished for flights arriving at a number of the busiest airports in the United States in the Traffic Management Advisor system (Swenson et al., 1997). This capability is being expanded to regional or national airspace, and for scheduling aircraft over any congested point in the NAS (Landry, Parley, & Hoang, 2005).

One proposal for increasing the capacity of the air traffic system is to allow such a system to negotiate "4D contracts," indicating navigational waypoints, altitudes, and times of arrival at those points for all aircraft. The flight management systems would then be responsible for adhering to the contract. These contracts would ensure conflict-free routes, and coordinate arrivals into congested resources, including airports.

Such a system would radically alter the roles of pilots and controllers in relation to their automation. Pilots and controllers would be monitors of a highly automated system, in comparison to their current roles of actively controlling the flight paths of aircraft. As such, this presents enormous challenges to the aviation HCI community.

Automating separation assurance. The primary responsibility of an air-traffic controller is to provide separation assurance between aircraft. Even if datalink and 4D contracts were established, the controller is still limited by the cognitively intensive process of monitoring for conflicts. This task is also under consideration for automated assistance, either on the flight deck or through a centralized system.

One proposal is to equip flight decks with a Cockpit Display of Traffic Information (CDTI) system, and utilize an advanced form of the current Traffic Collision and Avoidance System (TCAS). A second proposal is to utilize a centralized system for providing alerts (Erzberger, 2004). Such a system would automatically coordinate resolutions, and provide for an independent backup in case of failures to resolve. In both cases, automation would detect conflicts, identify resolutions, and (perhaps) automatically perform the resolutions. Such systems overcome the latencies in the human detection, communication, and action cycles; however, as the human operators are pushed back further from the actual operation of the systems, their ability to intervene in the case of automation failures wanes. Because of this, such automation must be highly reliable for the system to succeed.

UAVs in the Airspace System

The rapid development of uninhabited aerial vehicles (UAVs) will also greatly impact the air-traffic system of the future. To date such vehicles have been mostly constrained to special uses, such as the military. It seems likely, however, that these vehicles will need to be integrated into the air-traffic control system as their uses expand.

Recently, efforts have begun to identify the issues related to integrating UAVs into the airspace system (Wegerbauer, 2005). Such vehicles, while not piloted in the traditional sense, are controlled by a remote pilot. The issues related to their use of the airspace system concern collision danger and safe extraction of the vehicle in the case of a communication (or other system) failure. These are but the first issues to be dealt with, particularly as one considers more advanced UAVs with increasing amounts of autonomy.

CONCLUSIONS

Fitts (1951b) and his colleagues described the research challenges in visual displays as of 1951. The document is remarkable in that it spurred (or foresaw) a great deal of research that continues to this day. Some of that can be attributed to the general nature of the questions, but many of the specific research questions and methods have been examined and used over the last forty years. What was noticeably (and understandably) absent from the discussion of transitory displays was any reference to higher cognition. In some ways, this is indicative of the progress made in aerospace human-machine system research. Many of the difficulties associated with these concepts originated as higher levels (and greater amounts) of automation were added to the flight deck.

The progression of research has also followed this model. Early research resulted in a number of principles for integrating displays, for determining the form of displays for different purposes, and for the positioning of displays. Later work has concentrated on higher level cognitive issues such as mental models and situation awareness. In addition, work has been done on understanding how to apply automation, resulting in a number of principles regarding automation use and allocation of function. Most recently, new types of displays and automation such as warning systems and decision support systems have been introduced, yielding a new set of problems for researchers to tackle. As the air transportation system matures and the demand for ever-greater levels of capacity increases, it seems that greater reliance on automated systems will be required. For such a system to be successful, such automation must be based on sound principles.

References

Abbott, T. S., Mowen, G. C., Person, L. H., Jr., Keyser, G. L., Jr., Yenni, K. R., & Garren, J. F., Jr. (1980). *Flight investigation of cockpit-displayed traffic information utilizing coded symbology in an advanced operational environment*. Hampton, VA: NASA Langley Research Center.

Allen, R. B. (1983). Cognitive factors in the use of menus and trees: An experiment. *IEEE Journal on Selected Areas in Communication, 1*(2), 333–336.

Arthur, W. C., & McLaughlin, M. P. (1998). *User request evaluation tool (URET): Interfacility conflict probe performance assessment*. McLean, VA: MITRE.

Bainbridge, L. (1983). Increasing levels of automation can increase, rather than decrease, the problems of supporting the human operator. *Automatica, 19*, 775–779.

Bannon, L. J. (2000, August). *Understanding common information spaces in CSCW.* Workshop on Cooperative Organisation of Common Information Spaces, Delft, Technical University of Denmark.

Barco Orthogon. (2003). *Osyris*. Retrieved January 27, 2005, from http://www.barco.com/barcoview/downloads/BVW_Osyris_new_6p.pdf.

Bartlett, F. C. (1943). *Instrument controls and displays—Efficient human manipulation*. London, UK Medical Research Council.

Bejczy, A. K., & Paine, G. (1977). Displays for supervisory control of manipulators. In *MIT Proceedings of the 13th Annual Conference on Manual Control* (pp. 275–284), Cambridge, MA, Massachusetts Institute of Technology.

Beringer, D. B., Willeges, R. C., & Roscoe, S. N. (1975). The transition of experienced pilots to a frequency-separated aircraft altitude display. *Human Factors, 17*, 401–414.

Berndtsson, J., & Normark, M. (1999, November). The coordinative functions of flight strips: Air traffic control work revisited. *ACM Group99: International Conference on Supporting Group Work*, Phoenix, AZ.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer centred system*. San Francisco: Morgan Kaufman.

Billings, C. E. (1995). Situation awareness measurement and analysis: A commentary. D. Garland & M. Endsley (Ed.), *Proceedings of the International Conference on Experimental Analysis and Measurement of Situation Awareness*, Daytona Beach, FL, Embry Riddle University Press.

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*. Mahwah, NJ: Lawrence Erlbaum.

Brudnicki, D. J., & McFarland, A. L. (1997). *User request evaluation tool (URET) conflict probe performance and benefits assessment*. McLean, VA: MITRE.

Bryan, L. A., Stonecipher, J. W., & Aron, K. (1954). 180-degree turn experimen. *University of Illinois Bulletin, 54*(11), 1–52.

Chapanis, A. (1959). *Research techniques in human engineering*. Baltimore: The Johns Hopkins Press.

Civil Aeronautics Board (1953). *Civil Air Regulations* Part 4b. 4b.611 (b).

Civil Aeronautics Board (1957). *Civil Air Regulations* Part 4b. Amendment 4b-7.

Cohen, E., & Follert, R. L. (1970). Accuracy of interpolation between scale graduations. *Human Factors, 31*(5), 481–483.

Cole, E. L., Milton, J. L., & McIntosh, B. B. (1954). Eye fixations of aircraft pilots. IX. Routine maneuvers under day and night conditions, using an experimental panel, (Tech. Rep. No. 53-220) Wright Patterson AFB, OH, USAF WADC.

Conrad, R. (1951). Speed and load stress in sensory-motor skill. *British Journal of Industrial Medicine, 8*, 1–7.

Cooper, G. E. (1977). *A survey of the status of and philosophies relating to cockpit warning systems*. Moffett Field, CA: NASA Ames Research Center.

Cotton, B. (2003, July). *For spacious skies—the twenty-first century promise of free flight*. Paper presented at AIAA International Air and Space Symposium and Exhibition: The next 100 years. Dayton, OH.

Craik, K. (1943). *The nature of explanation*. Cambridge: Cambridge University Press.

Crane, P. M. (1992). Theories of expertise as models for understanding situation awareness. (Tech. Rep. No. ADP006943) Armstrong Lab: Williams, AZ.

Dashevsky, S. G. (1964). Check-reading accuracy as a function of pointer alignment, patterning, and viewing angle. *Journal of Applied Psychology, 48*, 344–347.

Davenport, W. W. (1978). *Gyro! The life and times of Lawrence Sperry*. New York: Charles Scribner.

Dearden, A., Harrison, M., & Wright, P. (1998). Allocation of function: scenarios, context and the economics of effort. *International Journal of Human Computer Studies, 52*(2), 289–318.

Dorris, A. L., Connolly, T., Sadosky, T. L., & Burroughs, M. (1977). More information or more data? Some experimental findings. *Proceedings of the Human Factors Society 21st Annual Meeting*, Santa Monica, CA: Human Factors Society.

Elkin, E. H. (1959). *Effect of scale shape, exposure time and display complexity on scale reading efficiency*. (Tech. Rep. No. 58-472) Wright Patterson AFB, OH, USAF WADC.

Endsley, M. R. (1988, October). *Design and evaluation for situation awareness enhancement*. Paper presented at the Human Factors Society 32nd Annual Meeting, Santa Monica, CA.

Endsley, M. R., & Rodgers M. D. (1996, September). *Attention distribution and situation awareness in air traffic control*. Paper presented at the 40th Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica, CA.

Eriksen, C. W., & Eriksen, B. A. (1974). Effects of noise letter upon the identification of a target letter ina nonsearch task. *Perception and Psychophysics, 16*, 143–149.

Erzberger, H. (2004, August). *Transforming the NAS: The next generation air traffic control system.* Paper presented at the 24th International Congress of the Aeronautical Sciences, Yokohama, Japan.

Erzberger, H., & Nedell, W. (1989). *Design of automated system for management of arrival traffic*. Moffett Field, CA: National Aeronautics and Space Administration.

Farber, E., & Paley, M. (1993, April). *Using freeway traffic data to estimate the effectiveness of rear-end collision countermeasures*. Paper presented at the Third Annual IVHS America Meeting, Washington, DC.

Ferrell, W. R., & Sheridan, T. B. (1967). Supervisory control of remote manipulation. *IEEE Spectrum, 4*(10), 81–88.

Fitts, P. M. (1951a). Engineering psychology in equipment design.SS. Sevens (Ed.), *Handbook of experimental psychology*. New York: Wiley.

Fitts, P. M. (1951b). *Human engineering for an effective air-navigation and traffic-control system*. Washington, DC: National Research Council Committee on Aviation Psychology.

Fitts, P. M., Jones, R. E., & Milton, J. L. (1950). Eye movements of aircraft pilots during instrument-landing approaches. *Aeronautical Engineering Review, 9*(2), 1–6.

Flach, J. M. (1995). Situation awareness: Proceed with caution. *Human Factors*, *37*, 149–157.

Fogel, L. (1959). A new concept: The kinalog display system. *Human Factors, 1*(1), 30–37.

Fracker, M. L. (1991). Measures of situation awareness: Review and future directions. (Tech. Rep. No. 1991-0127) Wright-Patterson AFB, OH, USAF Armstrong Laboratory.

Francis, G. (2000). Designing multifunction displays: An optimization approach. International *Journal of Cognitive Ergonomics, 4*(2), 107–124.

Gentner, D., & Stevens, A. L. (1983). *Mental models*. Hillsdale, NJ: Lawrence Erlbaum.

Gilbert, G. A. (1973). Historical development of the air traffic control system. *IEEE Transactions on Communications, 21*(5), 364–375.

Gosling, G. D. (2002, January). *Analysis of factors affecting the occurrence and severity of air traffic control operational errors*. Paper presented at the 2002 TCB Annual meeting, Washington, DC.

Green, P. (1984). *Driver understanding of fuel and engine gauges*. Warrendale, PA: Society of Automotive Engineers.

Hacker, E. W. (1984). Learning from the past: A fighter pilot's obligation. Retrieved July 29, 2005, from http://www.globalsecurity.org/military/library/report/1984/HEW.htm

Hanson, R. H., Payne, D. G., Shively, R. J., & Kantowitz, B. H. (1981). *Process control simulation research in monitoring analog and digital displays*. Paper presented at the Human Factors Society 25th Annual Meeting, Rochester, NY.

Harper, R., & Hughes, J. (1991). What a F-ing system! Send 'em all to the same place and then expect us to stop 'em hitting: Making technology work in air traffic control. In G. Burton (Ed.), *Technology in working order: studies of work, interaction and technology.* Cambridge, England, Rank Xerox Cambridge EuroPARC.

Heglin, H. J. (1973). *NAVSHIPS display illumination design guide: Human factors*. San Diego: Naval Electronics Laboratory Center.

Hersey, M. D. (1923). Aeronautic instruments, Section I: General classification of instruments and problems including bibliography. (Tech. Rep. No. 105) Langley, VA, NASA.

Hodge, M. H., & Reid, S. R. (1971). The influence of similarity between relevant and irrelevant information upon a complex identification task. *Perception and Psychophysics,* 13, 193–196.

Hughes, J. A., Randall, D., & Shapiro, D. (1992). Faltering from ethnography to design. In *Proceedings of The ACM Conference on Com-*

puter Supported Cooperative Work, Toronto, Ontario, ACM Press, pp. 115–122.

Iwasaki, S. (1996, April). Speeded digit identification under impaired perceptual awareness. Presented at Toward a Science of Consciousness 1996. Tucson, AZ, University of Arizona.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA: Harvard University Press.

Jones, D. G., & Endsley, M. R. (2000, October). *Can real-time probes provide a valid measure of situation awareness?* Paper presented at Human Performance, Situation Awareness and Automation: User Centered Design for the New Millennium Conference, Savannah, GA.

Knorr, D. (2003, December). *Free flight program performance metrics to date*. Washington, DC: Federal Aviation Administration.

Kuchar, J. K. (1996). Methodology for alerting-system performance evaluation. *Journal of Guidance, Control, and Dynamics, 19*, 438–444.

Landry, S. J., Farley, T., & Hoang, T. (2005, June). *Expanding the use of time-based metering: Multi-center traffic management advisor.* Paper presented at the 6th USA/Europe ATM 2005 R&D Seminar, Baltimore.

Lee, K. K., & Davis, T. J. (1996). The development of the final approach spacing tool (FAST): A cooperative controller-engineer design approach. *Journal of Control Engineering Practice, 4*(8), 1161–1168.

Lee, K. K., & Sanford, B. D. (1998). *Human factors assessment: The passive final approach spacing tool (pFAST) operational evaluation*. Moffett Field, CA: National Aeronautics and Space Administration.

Levy, J. L., Foyle, D. C., & McCann, R. S. (1998, October). *Performance benefits with scene-linked HUD symbology: An attentional phenomenon?* Paper presented at the 42nd Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica, CA.

Ljungberg, M., & Lucas, A. (1992, September). *The OASIS air traffic management system*. Paper presented at the 2nd Pacific Rim International Conference on Artificial Intelligence, Seoul, Korea.

McGreevy, M. W., & Ellis, S. R. (1986). The effect of perspective geometry on judged direction in spatial information instruments. *Human Factors, 28*, 439–456.

McRuer, D., & Graham, D. (1965). Human pilot dynamics in compensatory systems. (Tech. Rep. No. 65-5) Air Force Flight. Dynamics Lab, Wright-Patterson AFB, OH.

Mendel, M. B., & Sheridan, T. B. (1986). *Optimal combination of information from multiple sources*. Arlington, VA: Office of Naval Research.

Midkiff, A. H., & Hansman, R. J. (1993). Identification of important "party line" information elements and implications for situational awareness in the datalink environment. *Air Traffic Control Quarterly, 1*(1), 5–30.

Monty, R. A. (1973). Keeping track of sequential events: Implications for the design of displays. *Ergonomics, 16*, 443–454.

Mountford, S. J., & Somberg, B. (1981). Potential uses of two types of stereographic display systems in the airborne fire control environment. *Proceedings of the Human Factors Society, 25*, 235–239.

Murphy, M. R., McGee, L. A., Paler, E. A., Paulk, C. H., & Wempe, T. E. (1978). Simulator evaluation of three situation and guidance displays for V/STOL aircraft zero-zero landing approaches. *IEEE Transactions on Systems, Man, and Cybernetics, 8*, 19–29.

NavCanada. (2003). *SASS: Sequencing and scheduling system*. Retrieved January 27, 2005, from http://www.navcanada.ca/contentdefinitionfiles/TechnologySolutions/products/StandAlone/sass/SASSen.pdf.

Norman, D. A. (1993). *Things that make us smart*. Reading, MA: Addison-Wesley.

Olmos, O., Wickens, C. D., & Chudy, A. (1997, April). *Tactical displays for combat awareness: An examination of dimensionality and frame of reference concepts, and the application of cognitive engineering*. Paper presented at the 9th International Syposium on Aviation Psychology, Columbus, OH, Ohio State University.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors, 39*(2), 230–253.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics, 30*(3), 286–297.

Prevot, T., Shelden, S., Mercer, J., Kopardekar, P., Palmer, E., & Battiste, V. (2003, October). *ATM concept integrating trajectory-orientation and airborne separation assistance in the presence of time-based traffic flow management*. Paper presented at the 22nd Digital Avionics Systems Conference, Indianapolis, IN.

Pritchett, A. R. (2001). Reviewing the role of cockpit alerting systems. *Human Factors in Aerospace Safety, 1*(1), 5–38.

Pritchett, A. R., Hansman, R. J., & Johnson, E. N. (1995). Use of testable responses for performance-based measurement of situation awareness. In D. J. Garland & M. Endsley (Eds.), *Experimental analysis and measurement of situation awareness*. Daytona Beach, FL: Embry-Riddle Press.

Pulat, B. M., & Ayoub, M. A. (1979). A computer-aided instrument panel design procedure. *Proceedings of the Human Factors Society, 23*, 191–192.

Pylyshyn, Z. (2003). Mental imagery: In search of a theory. *Trends in Cognitive Science, 7*(3), 113–118.

Regal, D. M., Rogers, W. H., & Boucek, G. P. (1988). *Situational awareness in the commercial flight deck—definition, measurement, and enhancement*. Warrendale, PA: Society of Automotive Engineers.

Robinson, J. E. III, Davis, T. J., & Isaacson, D. R. (1997, August). *Fuzzy reasoning-based sequencing of arrival aircraft in the terminal area*. Paper presented at the AIAA Guidance, Navigation and Control Conference, New Orleans, LA.

Roscoe, S. N. (1968). Airborne displays for flight and navigation. *Human Factors, 10*(4), 321–332.

Roscoe, S. N. (1981). *Aviation psychology*. Ames, Iowa: The Iowa State University Press.

Roscoe, S. N. (1997). Horizon control reversals and the graveyard spiral. *CSERIAC Gateway 7*(3), 1–4.

Sabeh, R., Jorve, W. R., & Vanderplas, J. M. (1958). Shape coding of aircraft instrument zone markings. (Tech. Rep. no 57-260) Wright-Patterson AFB, USAF WADC.

Sanders, M. S., & McCormick, E. J. (1993). *Human factors in engineering and design*. New York: McGraw-Hill.

Sarter, N. B., & Woods, D. D. (1991). Situation awareness: A critical but ill-defined phenomenon. *The International Journal of Aviation Psychology, 1*(1), 45–57.

Scallen, S. F., Hancock, P. A., & Dudley, J. A. (1996). Pilot performance and preference for short cycles of automation in adaptive function allocation. *Applied Ergonomics, 26*(6), 397–403.

Schmidt, H. P., & Saint, S. P. (1969, December). *Human bottleneck in air traffic control*. Paper presented at the ASME Winter Annual Meeting, Los Angeles.

Schmidt, K., & Bannon, L. (1992). Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work, 1*(1), 7–40.

Schneiderman, B. (1998). *Designing the user interface.* Cambridge, MA: Addison-Wesley.

Seidler, K., & Wickens, C. D. (1992). Distance and organization in multifunction displays. *Human Factors, 34*, 555–569.

Selcon, S. J., & Taylor, R. M. (1989). Evaluation of the situational awareness rating technique (SART) as a tool for aircrew systems design. In *Proceedings of the NATO AGARD Conference on Situational Awareness in Aerospace Operations* (AGARD-CP-478). Springfield, VA: National Technical Information Service.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control.* Cambridge, MA: The MIT Press.

Sheridan, T. B. (1998). Allocating functions rationally between humans and machines. *Ergonomics in Design, 6*(3), 20–25.

Simmonds, G. R., Galer, M., & Baines, A. (1981). *Ergonomics of electronic displays*. Warrendale, PA: Society of Automotive Engineers.

Smith, K., & Hancock, P. A. (1995). Situation awareness is adaptive, externally directed consciousness. *Human Factors, 37*(1), 137–148.

Sokkappa, B. G. (1989). *The impact of metering methods on airport throughput*. McLean, VA: MITRE.

Starr, S. L., & Griesemer, J. R. (1989). Institutional ecology, "translations," and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Soology. *Social Studies of Science, 19*, 387–420.

Suchman, L. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge, UK: Cambridge University Press.

Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., Sanford, B., et al. (1997 June). *Design and operational evaluation of the traffic management advisor at the Fort Worth air route traffic control center*. Paper presented at the 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France.

Taylor, R. M. (1989). Situational awareness rating technique (SART): The development of a tool for aircrew systems design. In *Proceedings of the NATO AGARD Conference on Situational Awareness in Aerospace Operations* (AGARD-CP-478). Springfield, VA: National Technical Information Service.

Tipper, S. P., & Driver, J. (1988). Negative priming between pictures and words in a selective attention task: evidence for semantic processing of ignored stimuli. *Memory and Cognition, 16*(1), 64–70.

Vandevenne, H. F., & Andrews, J. W. (1993). Effects of metering precision and terminal controllability on runway throughput. *Air Traffic Control Quarterly, 1*(3), 277–297.

VanRullen, R., & Koch, C. (2003). Competition and selection during visual processing of natural scenes and objects. *Journal of Vision, 3*(1), 75–85.

VanRullen, R., & Thorpe, S. J. (2001). The time course of visual processing: from early perception to decision-making. *Journal of Cognitive Neuroscience, 13*(4), 454–461.

Veitengruber, J. E. (1977). Design criteria for aircraft warning, caution, and advisory alerting systems. *Journal of Aircraft, 15*(9), 574–581.

Ververs, M. P., & Wickens, C. D. (1998). Head-up displays: Effects of clutter, symbology, intensity, and display location on pilot performance. *The International Journal of Aviation Psychology, 8*(4), 377–403.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics, 22*, 589–606.

Vidulich, M. A. (1992, October). *Measuring situation awareness.* Paper presented at the Human Factors Society 36th Annual Meeting, Atlanta, GA.

Wegerbauer, C. (2005 May/June). The Access 5 project—Enabling direct routine UAS operations in the U.S. National Airspace System. *Unmanned Vehicles Magazine, 10*(3).

Well, A. (1971). The influence of irrelevant information on speeded classification tasks. *Perception and Psychophysics, 13*, 79–84.

Whitehurst, H. O. (1982). Screening designs used to estimate the relative effects of display factors on dial reading. *Human Factors, 24*(3), 301–310.

Whitfield, D., & Jackson, A. (1983). The air traffic controller's picture as an example of a mental model. In *Proceedings of the IFAC Conference on Analysis, Design, and Evaluation of Man-Machine System G. Johannsen & J. E. Rijnskorp* (Eds.) London: Pergamon.

Wickens, C. (2003). Aviation displays. In P. Tsang & M. Vidulich (Eds.), *Principles and practice of aviation psychology* (pp. 147–199). Mahwah, NJ: Lawrence Erlbaum.

Wickens, C. D., & Carswell, C. M. (1995). The proximity compatibility principle: Its psychological foundation and its relevance to display design. *Human Factors, 37*(3), 473–494.

Wickens, C. D., & Prevett, T. (1995). Exploring the dimensions of egocentricity in aircraft navigation displays. *Journal of Experimental Psychology: Applied, 1*(2), 110–135.

Wickens, C. D., Fadden, S., Merwin, D., & Ververs, P. M. (1998, November). *Cognitive factors in aviation display design*. Paper presented at the 17th Digital Avionics Systems Conference, Belleview, WA.

Wickens, C. D., Mavor, A. S., Parasuraman, R., & McGee, J. P. (1998). *The future of air traffic control: Human operators and automation*. Washington, DC: National Academy Press.

Wiener, E. L., & Curry, R. E. (1980). Flight-deck automation: Promises and problems. *Ergonomics, 23*(10), 995–1011.

Wittgenstein, L. (1922). *Tractatus logico-philosophicus*. London: Routledge & Kegan Paul.

Woods, D. D. (1984). Visual momentum: A concept to improve the cognitive coupling of person and computer. *International Journal of Man-Machine Studies, 21*, 229–244.

Wright, P., Dearden, A., & Fields, B. (1998). Function allocation: A perspective from studies of work practice. I*nternational Journal of Human Computer Studies, 52*(2), 335–355.

Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87–122.

·37·

USER-CENTERED DESIGN IN GAMES

Randy J. Pagulayan, Kevin Keeker, Thomas Fuller, Dennis Wixon and Ramon L. Romero
Microsoft Game Studios

Introduction **742**
Why Games Are Important **742**
Players Are as Diverse as the Games That They Play 742
Defining Games **742**
The Goals Are Different 742
The Rules Are in the Game World 743
Industry Competition Is Intense 743
Games Must Challenge 743
Types of Games **743**
Platforms 743
PC 743
Console 743
Portable devices 744
Cross platform 744
Game Types 744
Retail games (NPD classifications) 744
Casual games 744
Principles and Challenges of Game Design **745**
Identifying the Right Kind of Challenges 745
Addressing Different Skill Levels 745
Players Must Be Rewarded Appropriately 746
Collecting and Completing 746
Story 746
Technological Innovation Drives Design 747
Perceptual-Motor Skill Requirements 747
Balancing Multiplayer Games 747
Enabling Social Networking 748
Important Factors in Game Testing and Evaluation **748**
Game Designer Intent 748
Ease of Use 749
Starting a game 749
Tutorials or instructional gameplay 749
In-game interfaces 749
Mapping input devices to functions 749
Challenge 750
Pace 750
Summary 750
User Research in Games **751**
Introduction to Methods—Principles in Practice 751
Usability techniques 751
Structured usability test 751
Rapid iterative testing and evaluation method (RITE) 752
Variations on usability methods 753
Survey Techniques 754
Formative Playtest surveys 754
Development of the Playtest surveys 755
Type of Formative Studies 755
Critical facet Playtest 755
Initial experience Playtest 756
Extended Playtests 756
Focus groups 756
A Combined Approach 757
Conclusion **757**
Acknowledgments **758**
References **758**

INTRODUCTION

The intent of this chapter is to review principles and challenges in game design improvement and evaluation and to discuss user-centered techniques that address those challenges. First, we will present why games are important, followed by the definitions and differences between games and productivity software. In the next section, we discuss the principles and challenges that are unique to the design of games. That discussion provides a framework for what we believe include the core variables that should be measured to aid in game design testing and evaluation. The chapter will conclude with some examples of how those variables can be operationalized through methods used by the Games User Research group at Microsoft Game Studios.

WHY GAMES ARE IMPORTANT

Computers that appeared commercially in the 1950s created a technological barrier that was not easy to overcome for the greater population. Only scientists, engineers, and highly technical persons were able to use these machines (Preece, Rogers, Sharp, Benyon, Holland, & Carey, 1994). As computers became less expensive, more advanced, and more reliable, the technology that was once available only to a small group of people was now permeating itself throughout the entire population and into everyday life. In order to ease this transition, the need for a well-designed interface between the user and the technology became a necessity. Computer games come from similar origins, and potentially may head down a similar path. Some early attempts at making commercial video games failed due to unnecessarily high levels of complexity. Nolan Bushnell, cofounder of Atari, stated the issue quite succinctly: "No one wants to read an encyclopedia to play a game" (Kent, 2000, p.28). In retrospect, some of the most successful video games were the ones that were indeed very simple.

Entertainment in general is a field of great importance to everyone. Today, the video games industry is one of the fastest growing forms of entertainment to date. According to the Entertainment Software Association (ESA), revenue from computer and console games has grown from 3.2 billion dollars in 1994 (Interactive Digital Software Association, 2000) to 7.3 billion dollars in 2004 (Entertainment Software Association, 2005). Approximately 248 million games were sold in 2004 (Entertainment Software Association, 2005). To put this into perspective, according to the ESA, 47% of Americans either purchased or planned to purchase one or more games in 2005 (Entertainment Software Association, 2005). These statistics do not even consider the international importance of video games.

Video games are quite important to the software development industry due to their roles in pushing research and technology forward. Games have acted as catalysts for hardware innovations (e.g., graphics cards, processors, and sound cards; e.g., Kroll, 2000), research on the relationships between artificial intelligence and synthetic characters or computer graphics (e.g., Funge, 2000; Laird, 2001), and physics (e.g., Hecker, 2000). These are all areas that previously were confined to engineering and computer science communities.

Players Are as Diverse as the Games That They Play

With annual sales reaching 8 billion dollars across a broad range of game genres and platforms, it is no surprise that the players are also quite diverse. Video games are no longer associated with male adolescents and young adults. According to the ESA (2005), 75% of American heads of household, over half of all Americans over the age of 9 have played, and 19% of Americans over the age of 50 years old have played an electronic game. The average age of those who play video games is 30, and 43% of Americans who play games are female (ESA, 2005). In addition, the extremely popular casual online games market is dominated by older females.

Assumptions that games are limited to a narrow market may stem from claims that the electronics industry has traditionally been marketed to boys (e.g., Chaika, 1996). Controversial beliefs about males possessing an innate proclivity towards video gaming or an innate skill advantage over females are commonly sported in casual conversation and sometimes offered by scholarly publications (e.g., Williams, 1987). While Nintendo's positioning has been demonstrated by previous marketing campaigns and promotions on such things as cereal boxes (Herz, 1997; Provenzo, 1991), genetic claims tend to be driven by speculation and stereotype.

DEFINING GAMES

As stated, the gaming industry is becoming as widespread and popular as computers and televisions. As the industry matures, games become more complex, making games difficult to define narrowly. Articulating a clear and succinct set of principles that capture the essence of games is not straightforward. However, shedding some light on the difference between games and productivity applications may help. When comparing games to productivity software, principles and methods can be successfully applied to both games and productivity applications. Regardless of the domain, the same techniques would be applied to understand a misleading button label or a confusing process model.

At the same time, there are fundamental differences between productivity applications and gaming applications. Some of the differences can be clear. Games are usually more colorful, wacky, and escapist than productivity applications, with the inclusion of interesting story lines, or animated characters.

User-centered design principles have not reached game makers to the degree that they have influenced other electronic applications. The following sections will describe some differences that have important implications for user research on games.

The Goals Are Different

At their roots, productivity applications are tools. Similarly, the design intentions behind productivity applications are to make task performance easier, quicker, less error prone, more quality driven, and accessible to populations of interest. In this sense, a word processor or spreadsheet differs only from a powered woodworking tool in terms of its complexity and domain

of application. One can argue that better word processors allow a wider set of people to make documents better, faster, and easier and with fewer errors. Similarly, a power mitre box cuts wood with angles that are more precise, faster, and more reliable than a hand mitre box or a hand saw. The focus of design and usability is to produce an improved product or result.

Games at their roots are different. Games are intended to be a pleasure to play. Ultimately, games are like movies, literature, and other forms of entertainment. They exist in order to stimulate thinking and feeling. Their outcomes are more experiential than tangible. This is not to say that word processors or other tools cannot be a pleasure to use or that people do not think or feel when using them, but that is rarely their primary design intention. In a good game, both the outcomes and the journeys are rewarding. This fundamental difference leads us to devote more of our effort to measuring perceptions, while productivity applications focus more on task completion.

The Rules Are in the Game World

Games create an artificial world with artificial objectives. Capturing the king has a chess-specific meaning that is different from the precise meaning outside of the chess world. In some sense, this simplifies the problem of the game designer and of the usability practitioner. User goals are almost completely defined by the game designer. The resulting concern is to ensure that user interpretations match the game design's expectations (or differ in a creative and entertaining manner).

Industry Competition Is Intense

Competition within the games industry is more intense than other software domains. Games compete with each other and many other forms of entertainment for your attention. Pretend that you have to write a book chapter. For that, there are a limited number of viable tools: handwriting, dictation, a typewriter, or a limited set of software applications. However, imagine that you have a few hours to kill and that you want to avoid productivity at all costs. You could consult the *New York Times* crossword puzzle, argue politics with your best friend, watch a mystery film, or stock up with health potions and attack demons in *Diablo* (2000).

Games Must Challenge

Perhaps the most important distinction between games and productivity applications is the complex relationship between challenge, consistency, and frustration. To illustrate this difference, consider the following example. It has been said that the easiest game to use would consist of one button labeled "Push." When you push it, the display would say, "YOU WIN." This game would have few, if any usability issues. However, it would not be fun either. Games deliberately impose constraints, whereas productivity applications attempt to remove them. Games deliberately evolve, whereas productivity applications strive for consistency. At their hearts, the fun in most games comes from learning how to overcome ever-increasing challenges. Ironically, some educators, practitioners, and researchers think it comes from fantasy and bright colors.

Many of the characteristics mentioned above are not unique to games. All of these characteristics are relevant to the discussion of the ways in which user research must adapt to be useful for game development. Later we will discuss how these differences create particular challenges for both the design and evaluation of games.

TYPES OF GAMES

In the following sections, we review some common gaming platforms and game types.

Platforms

One of the simplest classifications of games is by the platform or hardware on which they are played. Cassell and Jenkins (2000) differentiated between games played on a PC from those played on a console. Crawford (1982) divided games into even finer categories; arcade/coin operated, console, PC, mainframe, and handheld. Different gaming platforms can be differentiated by their technical capabilities, physical control interface, visual interface, and the context in which they are played.

PC. A useful distinction can be made between PC games that are normally acquired through retail outlets on CDs, persistent world games in which much of the content lives on the Internet, and casual, web-based games. While there are many technical differences between these kinds of games, the important distinction is the business model and level of investment required by the players. Most retail games need to rely heavily on flash, reputation, and recognizable novelty to attract users to a relatively large investment. Continued investment in the game is useful only to build a reputation or to convince customers with higher thresholds that they will get their moneys' worth from the game. Massively multiplayer online persistent worlds (MMOs) require continued user investment to obtain monthly subscription fees. The long learning curves and reinforcement schedules in MMOs are tailored to make players invest more time, money, and effort for long-term rewards. Many casual gamers are attracted to free, familiar games. For these casual users, gameplay is squeezed between activities that are more important to them. As a result, games with very little learning investment are very appealing to casual gamers. Removing penalties for setting aside the game (for minutes, hours, days, or forever) can mean the difference between successful and unsuccessful casual games (see the casual games section for more details).

Console. For user research purposes, the most important unique characteristic of console video game systems is the advantage of a fixed set of hardware. By contrast with PC games, there is very little game setup, minimal maintenance efforts (e.g., few software patches, video card conflicts, etc.), and a consistent set of input device capabilities. Some game genres are

more popular on console than PC. This difference in popularity can usually be attributed to differences in the input devices that are typically associated with each platform. It is hard to enter text using a gamepad. Most popular MMOs are still published for PCs in spite of the fact that fast Internet connections and hard drives are becoming more popular additions to console systems. This may change as it becomes easier to attach other devices such as keyboards or voice-command hardware that would make communication more robust.

More recently, casual games are also available for all console systems, including Xbox 360, where consumers can download, try, and purchase casual games directly from the console via a broadband connection.

Portable devices. Portable devices include portable gaming systems (handhelds), mobile phones, and personal digital assistants (PDAs). Handheld devices are a standardized platform, with screens and controls designed primarily for gaming. Mobile phones on the other hand provide greater challenges to creating meaningful gameplay experience, including limited screen resolution and small controls (Hyman, 2005). Mobile games generally must support games that can be played in shorter blocks of time and that can work well with the limitations of the handsets. The unlimited combination of handset controls, processor and network capabilities, and runtime environments create even more challenges to developing quality games (IDGA, 2005; Hyman 2005).

Consumers primarily view mobile phones as communication devices, which are not meant to replace the gaming experience available on handhelds such as the Nintendo DS and Sony PSP (Moore & Rutter, 2004). Unlike handheld gaming devices, mobile phones are generally carried at all times. This creates an opportunity to provide gaming at any time, anywhere (Hymen, 2005; IDGA, 2005). However, with mobility comes the greater need to design for time, location, social, emotional, and motivational context (Sidel & Mayhew, 2003).

Cross platform. Cross-platform convergence is an emerging trend in the games industry that complicates the classification of games by platform and that creates new challenges in user-centered game design. Developers are now creating multiple versions of a game that interact with one another across platforms. For example, Nintendo games such as *Animal Crossing* (2002), *The Legend of Zelda: The Wind Waker* (2003), and *Harvest Moon* (2003) allow users to connect a Game Boy portable device to the GameCube console to access additional features and content or to use the Nintendo DS as a touch screen controller. Another example of cross-platform games would be a title where the player may execute specific planning or strategic tasks on a mobile device at any time, then have those actions affect the full game experience the next time it is played on a console or PC. This cross-platform design effect has been referred to as a reciprocation effect (Yuen, 2005).

The next generation of cross-platform experiences may involve matching players for real time or asynchronous competitive and co-operate multiplayer experiences regardless of what platform they are on (portable, PC, console, etc). As these trends become realized, the promise of interesting challenges and opportunities for game developers continue to increase.

Game Types

Retail games (NPD classifications). The NPD Group (a marketing company) uses a fine-grained classification scheme for game type that is referred to quite often in the games industry. They offer the following classes and subclasses as seen in Table 37.1. Games can also be categorized in a more granular fashion by splitting each genre up into a subgenre. For instance, action games can be divided into Action Driving Hybrid (*True Crime, GTA3,* etc.), Action Combat (*Max Payne, Tomb Raider*, etc.), and Platformer (*Jak and Daxter, Super Mario Sunshine*, etc). In general, the genres represent types of gameplay mechanics, themes, or content.

Casual games. Casual games are a rapidly growing segment of the video games market, producing an estimated $500 million in online sales in 2005 (IDG Entertainment, 2005). Casual games as a whole are appealing to a broad range of users, can be found on all platforms, and represent several genres.

TABLE 37.1 The NPD Group Super Genre Classification Scheme for Games

Category	Description
Action	Control a character and achieve one or more objectives with that character.
Fighting	Defeat an opponent in virtual physical hand-to-hand or short-range combat.
Racing	Complete a course before others do and/or accumulate more points than others while completing a course.
Shooter	Goal is to defeat enemies in combat with ranged weapons (first-person shooters, third-person shooters).
Strategy	Strategically manage resources to develop and control a complex system in order to defeat an opponent or achieve a goal. The goal is to defeat opponent(s) using a large and sophisticated array of elements.
Role-Playing	Control a character that is assuming a particular role in order to achieve a goal or mission. Rich story and character development are common.
Family entertainment	The primary objective is to interact with others and/or to solve problems. This genre includes puzzles and parlor games.
Children's entertainment	Same as family entertainment, but geared to a younger audience.
Sports	Manage a team or control players, to either win a game or develop a team to championship status. Involves individual, team, and extreme sports,
Adventure	Control a character to complete a series of puzzles that are required to achieve a final goal.
Arcade	Games on coin-operated arcade machines, or games that have similar qualities to classic arcade games. Generally, they are fast paced, action games.
Flight	Plan flights and pilot an aircraft in a realistic, simulated environment
All other games	Educational, compilations, non flight simulators, rhythm games, etc.

For these reasons, they do not fit into either of the previous categorization schema. Casual games generally meet the following requirements: (a) are easy to start and control; (b) can be enjoyed in small time intervals; (c) do not require an extensive investment in time to enjoy or progress, and therefore usually do not have deep linear story lines; and (d) have discreet goals and rules that are intuitive or easy to learn. Casual games often have a significantly lower development costs, can be distributed digitally (or via physical media), offer trial versions and lower retail prices (or are free), and utilize lower system requirements and storage space. These requirements and attributes make these games well suited for platforms such as PC distributed via the web and mobile devices, where they are most common.

Websites such as Real Arcade, Pogo.com, Yahoo! Games, and MSN Games provide free online casual games, pay per play casual games, subscription services, downloadable games that include free trials, and games that can be purchased for $5 to $20. Mobile phone carriers and a variety of websites provide casual games for mobile devices. Although there are feature rich, 3-D noncasual games available for mobile devices, the top selling titles are casual games such as *Tetris, Bejeweled, Bowling*, and *Pacman* (Hyman, 2005).

PRINCIPLES AND CHALLENGES OF GAME DESIGN

Having differentiated games from other applications, we can look at some of the unique issues in game design and evaluation.

Identifying the Right Kind of Challenges

Games are supposed to be challenging. This requires a clear understanding of the difference between good challenges and frustrating usability problems. Most productivity tools struggle to find a balance between providing a powerful enough tool set for the expert and a gradual enough learning curve for the novice. However, no designer consciously chooses to make a productivity tool more difficult. The ideal tool enables users to experience challenge only in terms of expressing their own creativity. For games, learning the goals, strategies, and tactics to succeed is part of the fun.

Unfortunately, it is not always clear which tasks should be intuitive (e.g., easy to use) and which ones should be challenging. Input from users becomes necessary to distinguish good challenges from incomprehensible design. Take a driving game for example. It is not fun having difficulty making your car move forward or turn. However, learning to drive is still a fundamental part of the challenge in the game. While all cars should use the same basic mechanisms, varying the ways that certain cars respond under certain circumstances may be fun. It should be challenging to identify the best car to use on an icy, oval track as opposed to a rally-racing track in the middle of the desert. The challenge level in a game must gradually increase in order to maintain the interest of the player.

Addressing Different Skill Levels

Unfortunately, not all players start from the same place in terms of gaming experience or talent. Obviously, frequent failure can be a turn off. Success that comes too easily can also become repetitive. Games must address the problem of meeting all players with the correct level of challenge. Tuning a game to the right challenge level is called "game balancing."

There are many ways to balance the difficulty of the game. The most obvious way is to let players choose the difficulty themselves. Many games offer the choice of an easy, medium, or hard difficulty level. While this seems like a simple solution, it is not simple to identify exactly how easy the easiest level should be. Players want to win, but they do not want to be patronized. Too easy is boring and too hard is unfair. Either experience can make a person cease playing.

Another approach to varying skill levels is to require explicit instruction that helps all users become skilled in the game. You might imagine a tutorial in which a professional golfer starts by explaining how to hit the ball and ends by giving instruction on how to shoot out of a sand trap onto a difficult putting green. Instruction, however, need not be presented in a tutorial. It could be as direct as automatically selecting the appropriate golf club to use in a particular situation with no input from the user, similar to the notion of an adaptive interface, where the interface provides the right information at the right time.

The environments, characters, and objects in a game provide another possibility for self-regulation. Most games will offer players some choices regarding their identities, their opponents, and their environments. The better games will provide a variety of choices that allow users to regulate the difficulty of their first experiences. With learning in mind, it is not uncommon for novice players to choose champion football teams to play against weak opponents. As long as players can distinguish the good teams from the bad ones, and the teams are balanced appropriately, users will be able to manage their own challenge levels.

Some games take it even further by identifying the skill level of the player and regulating the instruction-level appropriately. In this situation, instruction can be tuned to the skill level of the player by associating it with key behavioral indicators that signifies that the player is having difficulty. If the game does not detect a problem, it does not have to waste the player's time with those instructions. In *Halo 2* (2004), the game detects difficulties that a player may have with certain tasks. For example, to get into a vehicle, the player must press and hold the X button on his or her controller when standing next to the driver's seat of that vehicle. If the player is standing in the right position, but taps the X button repeatedly (instead of holding the button down), the game will present a more explicit instruction to press and hold the button. This is just one of many dynamic instructions that appear throughout *Halo 2* based on behavioral indicators of player difficulty.

Productivity tools have implemented similar problem-identification features, but often with mixed success due to the open nature of tasks in most productivity applications. Good game tutorials have succeeded by setting clear goals and completely constraining the environment. Doing so focuses the user on the specific skill and simplifies the detection of problem behavior. Other lessons from game tutorial design will be described in later sections of this chapter.

Another in-game approach to auto-regulating the difficulty level requires adjusting the actual challenge level of the opponents during the game. Evaluating the success of the player and adjusting the opponent difficulty during the game is often called "dynamic difficulty adjustment," or "rubber banding." When players perform very skillfully, their performances are moderated by computer-generated bad luck and enhanced opponent attributes. In a football game, the likelihood of fumbling, throwing an interception, or being sacked may increase as the player increases his or her lead over their opponent. Even though this may seem like a good solution, there can be a downside. Most people would prefer to play a competitive game (and win) than to constantly trounce a less-skilled opponent. However, overdeveloped rubber banding can cheat a skilled player out of the crucial feeling of mastery over the game.

A final approach focuses on providing tools that maximize the ability of the trailing player to catch up with the leading player. The key for the game designer is to think of ways to maintain challenge, reward, and progress for the unskilled player without severely hampering the skilled player. One interesting and explicit example of this is found in *Diddy Kong Racing* (1997). In this game, the racer can collect bananas along the roadway. Each banana increases the top speed of your car. The player can also collect missiles to fire forward at the leading cars. Each time you hit a car with a missile, it not only slows the car's progress, but it jars loose several bananas that the trailing player can pick up. Thus, trailing players have tools that they can use to catch the leaders even if the leaders are not making any driving mistakes. The chief distinction between this and dynamic difficulty adjustment is that the game is not modifying skills based on success. Instead, the rules of the game provide the trailing player with known advantages over the leader.

Players Must Be Rewarded Appropriately

Explicit or slow reinforcement schedules may cause users to lose motivation and quit playing a game. Because playing a game is voluntary, games need to quickly grab the user's attention and keep him or her motivated to come back repeatedly. One way to accomplish this is to reward players for continued play. Theories of positive reinforcement suggest behaviors that lead to positive consequences tend to be repeated. Thus, it makes sense that positive reinforcement can be closely tied to one's motivation to continue playing a game. However, it is less clear which types of reinforcement schedules are most effective.

Although the model should not necessarily be used for all games, research suggests that continuous reinforcement schedules can establish desired behaviors in the quickest amount of time (Steers & Porter, 1991). Unfortunately, once continuous reinforcement is removed, desired behaviors extinguish very quickly. Use of partial reinforcement schedules take longer to extinguish desired behaviors but may take too long to capture the interest of gamers. Research suggests that variable ratio schedules are the most effective in sustaining desired behaviors (Jablonsky & DeVries, 1972). This kind of schedule is a staple of casino gambling games, in which a reward is presented after a variable number of desired responses. Overall, there is no clear answer. Creating a game that establishes immediate and continued motivation to continue playing over long periods is a very complex issue.

Another facet of reinforcement systems that may impact enjoyment of a game is whether the player attributes the fact that they have been playing a game to extrinsic or intrinsic motivations. Intrinsic explanations for behavior postulate that the motivators to perform the behavior come from the personal needs and desires of the person performing the behavior, whereas extrinsically motivated behaviors are those that people perform in order to gain a reward from or please other people. In research on children's self-perceptions and motivations, M. Lepper, Greene, and Nisbett (1973) discovered that children who were given extrinsic rewards for drawing were less likely to continue drawing than those who had only an intrinsic desire to draw. The conclusion that they drew is that children perceived their motivations to draw as coming from extrinsic sources and thus discounted their self-perceptions that they liked to draw.

The same may be true of reward systems in games (Lepper, M. R. & Malone, 1987; Malone, 1981). To a certain degree, all reinforcement systems in games are extrinsic because they are created or enabled by game developers. Some reward systems are more obviously extrinsic than others are. For instance, imagine the following rewards that could be associated with combat in a fantasy role-playing game (RPG). The player who slays a dragon with the perfect combination of spell casting and swordplay may acquire the golden treasure that the dragon was hoarding. In this situation, the personal satisfaction comes from being powerful enough to win and smart enough to choose the correct tactics. The gold is an extrinsic motivator. The satisfaction is intrinsic. By analogy from M. Lepper, Greene, and Nisbett's (1973) research, feelings of being powerful and smart (intrinsic motivators) are more likely to keep people playing than extrinsic rewards.

Collecting and Completing

The chief goal of many games is to acquire all of the available items, rewards, and knowledge contained in the game. In games such as *Pokemon Crystal* (2000), the central challenge is to acquire as many of the *Pokemon* characters as you can and learn all of their skills well enough to outsmart your opponent at selecting the right characters for a head-to-head competition. Not coincidentally, the catch phrase for the *Pokemon Crystal* game is "Gotta catch 'em all!"

This game mechanic is also used by numerous games to add depth and repeat play. Though this is not the primary mechanic in *Madden NFL 06* (2005), the ability to collect electronic player cards and use them strategically in games provides incentive for gamers to experiment with much of the content that they may not experience if playing through a standard season.

Story

Characters and narrative help gamers attach meaning and significance to action sequences. Those in the games industry propose that many games neither have nor need a story. It is our contention that the key to understanding narrative in games is

to realize that story lines may be both embedded in the game, or they may emerge in the course of playing a game.

When most consumers think about story, they think about embedded story lines (Levine, 2001). *Final Fantasy X* (2001) tells its story by cutting action sequences with a series of full-motion cut scenes, real-time cut scenes, and player-driven character interactions. The embedded story forms a central part of the appeal of the game. Levine (2001) pointed out that much of the narrative in a game emerges in the successes and failures experienced by players throughout the course of the game. This is especially true of multiplayer games, in which story is often generated exclusively by the interactions of the participants. As Levine described it, the story is generated by replaying an abstract narrative structure with a strict set of rules and possibilities within a novel set of circumstances. Sporting events both within and outside of the video game world provide an excellent example of this type of narrative. No author scripted the result of the last World Cup tournament, but each such event has the potential to be an epic narrative for both participants and viewers.

Technological Innovation Drives Design

There is a great deal of pressure on designers to utilize new technologies that may break old interaction models. The desire to experience ever more realistic and imaginative games has pushed game developers into engineering and computer-science research domains. Likewise, technology often drives game design in order to showcase new capabilities. The constant demand for novelty can be strong enough incentive for game makers to try untried designs, "spruce up" familiar interfaces, and break rules of consistency. For example, the original *NFL2K* series (1999) sported a new interface model in which users selected interface areas by holding the thumbstick in a direction while pressing a button. It is possible that Sega chose the new design primarily because it was new and different from existing interfaces. It required the somewhat new (at that time) mechanics of the thumbstick on the controller, it minimized movement because one could point directly at any given item in the menu, and it was cleverly shaped like a football (which made more sense for *NFL2K* than for another sport). However, errors due to more error-prone targeting and the hold and click interaction metaphor made the system harder for first-time players to use.

Perceptual-Motor Skill Requirements

The way that functions are mapped onto available input devices can determine the success or failure of a game. A crucial part of the fun in many games comes from performing complex perceptual-motor tasks. While today's arcade-style games include increasingly more sophisticated strategic elements, a core element of many games is providing the ability to perform extremely dexterous, yet satisfying physical behaviors. These behaviors are usually quick and well-timed responses to changes and threats in the environment. If the controls are simple enough to master and the challenges increase at a reasonable difficulty, these mostly physical responses can be extremely satisfying (Csiksentmihalyi, 1990).

Problems can arise when games require unfamiliar input devices. This is a common complication in console game usability research because new console systems usually introduce new input device designs unique to that system (see Fig. 37.1). Furthermore, game designers often experiment with new methods for mapping the features of their games to the unique technical requirements and opportunities of new input devices. Unfortunately, this does not always result in a better gameplay experience.

Balancing Multiplayer Games

As we have seen, different strategies can be used to support a broad range of skill player skill levels. While this is complicated in single-player games, it becomes even more daunting when players of different skills make up both the opponents and the allies. By far the most common strategy for regulating the challenge level of online multiplayer games is to provide strong matchmaking tools. *Internet Backgammon* (2001) automatically matches players from around the world based on their self-reported skill level. Other games use algorithms that count player winning percentages and strength of opponent.

Another strategy seeks to solve skill balance problems outside of the game by offering players a wide array of arenas and game types. Rather than actively connecting players of like skill, this approach provides a broad array of places to play and allows players to self-select into game types that suit their style and attract the players with whom they want to associate. A final approach, used frequently by instant messaging clients (e.g., Yahoo IM, MSN Messenger), is to make it easy to start a game with friends. Though skill levels of friends may not always match, you are presumably less likely to perceive the match as unfair against people that you know.

Game designers also employ a variety of in-game strategies to balance out the competition. The most common way to allow less skilled players to compete effectively with players that are more skilled is to play team games. Many games allow players to



FIGURE 37.1. This is a sample of different types of input devices used in games; from the traditional keyboard and mouse, to different console gamepads.

take on a variety of roles. *Capture the Flag* is a common backyard tag-style game that forms the basis for a game type in many first-person shooters. Some players go on offense to capture the opponent's flag, some stay back to defend their flag. Likewise, some players may take a long-range sniper weapon to frustrate the enemy, while others take weapons that are more effective at close range. The defending and sniping roles can be more comfortable for some novice players because they allow the player to seek protective cover, they require less knowledge of the play field, and one-on-one combat can be avoided.

Some first-person shooters also allow the game host to set handicaps for successful players and bonuses for less successful players. For example, one version of *Unreal Tournament* increased the size of the player's character with every point that they won, and decreased the size of the character every time that the character died. The size differences made successful players easier targets and unsuccessful players more difficult targets.

Enabling Social Networking

Skill level is not the only dimension of importance. Recent systems, such as the Xbox Live service, seek to group players of similar attitudes and behaviors. The goal is to place more antagonistic players with like-minded others who appreciate tough talk, while protecting players who do not want to or should not be exposed to aggressive or offensive language.

This is just one of many tools that games have developed to promote group play and improve communication between online players. Most online games include some form of in-game messaging. Often this messaging is tailored for fast and efficient communication of key functions or timely game events. Most massively multiplayer online RPGs have negotiation systems that help people trade objects safely and efficiently. Most first-person shooters present automatic messages telling you which players have just been killed by whom. Real-time strategy (RTS) games allow gamers to set flares or markers on the map to notify your allies of key positions. Each game genre has a key set of in-game communication tools to support the game play.

More frequently, game designers are starting to embed cooperative tools into the environment itself. Much of the benefit of vehicles in first-person shooting games comes from their use as cooperative tools. Used in concert, these tools can often be extremely effective. From a game design perspective, they provide a great incentive for people to come together, strategize, and work co-operatively. These shared successes can be a huge part of the fun in online games.

Massively multiplayer games employ a wide variety of incentives for players to form groups. Success comes from taking on missions that are far too dangerous for any single player to accomplish on their own. Players choose a role and learn to complement the strengths of the other members of their group in order to overcome fearful enemies and accomplish great quests. Recent massively multiplayer games have invested even more in the creation of large-scale guilds. Most massively multiplayer games intentionally keep large areas of information secret from their players—to provide the need and the opportunity for symbiotic relationships between experts and novices. In addition, modern MMOs often provide pyramid-scheme style bonuses in wealth and experience to the leaders and captains of guilds. In return, members receive physical, material, educational, and social protection from their leaders and peers.

IMPORTANT FACTORS IN GAME TESTING AND EVALUATION

Most game genres are subtly different in the experiences that they provoke. It may seem obvious that the point of game design is making a fun game. Some games are so fun that people will travel thousands of miles and spend enormous amounts of money to participate in gaming events. However, we would like to propose a potentially controversial assertion. That the fundamental appeal of some games lies in their ability to challenge, to teach, to bring people together, or simply to experience unusual phenomena. Likewise, the definition of fun may be different for every person. When you play simulations games, your ultimate reward may be a combination of learning and mastery. When you play something like the *MTV Music Generator* (Codemasters, 1999), your ultimate reward is the creation of something new. When you go online to play card games with your uncle in Texas, you get to feel connected. *Flight Simulator 2000* (2000) lets people understand and simulate experiences that they always wished they could have. While these may be subcomponents of fun in many cases, there may be times when using "fun" as a synonym for overall quality will lead to underestimations of the quality of a game. While a fun game is often synonymous with a good game, researchers are warned to wisely consider which measures best suit the evaluation of each game that they evaluate.

Game Designer Intent

As stated earlier, the goal of a game is to create a pleasurable experience. Goals in a game are not necessarily derived from external user needs as in productivity applications. In games, goals are defined in accordance with the game designer's vision, which is a novel position because historically, success in has been defined by the accomplishment of user tasks and goals (Pagulayan, Gunn, Romero, 2006). When approaching a game for user-centered design or testing, it is best to assume the role of facilitating the designer's vision for the game (Pagulayan & Steury, 2004; Pagulayan, Steury, Fulton, & Romero, 2003) because many times, only the designer can recognize when the player experience is not being experienced as intended. In traditional usability testing, it is often very recognizable when there is user error, but not so in games.

Davis, Steury, and Pagulayan (2005) discussed a case study in the game *Brute Force* which revealed that players were not encountering certain gameplay features early enough in their gameplay experience. The players were not failing, but that players were taking much longer to play through the second mission than intended. By understanding the design vision, the authors were able to work with the designers to provide feedback that resulted in shortening the second mission and in reordering other missions to match the design intent of the game.

Ease of Use

The ease of use of a game's controls and interface is closely related to fun ratings for that game. Think of this factor as a gatekeeper on the fun of the game. If the user must struggle or cannot adequately translate their intentions into in-game behaviors, they will become frustrated. This frustration can lead the user to perceive the game as being unfair or simply inaccessible (or simply not fun). Thus, it becomes very clear why usability becomes very important in games. Ease of use should be evaluated with both usability and attitude-measurement methodologies, which are discussed later in the chapter.

Starting a game. Starting the kind of game that the user wants is an easily definable task with visible criteria for success. This is something one can measure in typical usability laboratories. Though designers often take game shell (the interface used to start the game) design for granted, a difficult or confusing game shell can limit users' discoveries of features and impede their progress towards enjoying the game. The immediate concern for users can be starting the kind of game that they want to play. Games often provide several modes of play. When the game shell is difficult to navigate, users may become frustrated before they have even begun the game. For example, we have found that many users are unable to use one of the common methods that sports console games use to assign a game controller to a particular team. This has resulted in many users mistakenly starting a computer versus computer game. Depending on the feedback in the in-game interface, users may think that they are playing when, in fact, the computer is playing against itself. In these cases users may even press buttons, develop incorrect theories about how to play the game, and become increasingly confused and frustrated with the game controls. The most effective way to avoid these problems is to identify key user tasks and test their usability.

Pagulayan, Steury, Fulton, et al. (2003) discussed a case study where they found issues with difficulty settings in the game shell. In early usability testing, participants were having problems with setting the difficulty level of opponents in *Combat Flight Simulator* (2000). This is a case where the users' gameplay experiences would have been quite frustrating because of a usability error in the game shell if it were not addressed.

Tutorials or instructional gameplay. As mentioned earlier, tutorials are sometimes necessary to introduce basic skills in order to play the game. In this situation, instructional goals are easily translated into the usability labs with comprehension tasks and error rates.

One of the risks of not testing tutorials or instructional missions is inappropriate pacing, which can often result from an ill-conceived learning curve at the start of the game. Many games simply start out at too difficult a challenge level. This is an easy and predictable trap for designers and development teams to fall into because when designers spend months (or even years) developing a game, they risk losing track of the skill level of the new player. A level that is challenging to the development team is likely to be daunting to the beginner.

Unfortunately, the reverse can also be troubling to the new user. Faced with the task of addressing new players, designers may resort to lengthy explanations. Frequently, developers will not budget time to build a ramped learning process into their game. Instead, they may realize late in the development cycle that they need to provide instruction. If this is done too abruptly, the learning process can end up being mostly explanation, and to be frank, explanation is boring. The last thing that you want to do is to bore your user with a longwinded explanation of what they are supposed to do when they get into your game. It is best to learn in context and at a measured pace or users may just quit the game.

A very positive example is the first level of *Banjo Kazooie* (2000). At the start, the player is forced to encounter a helpful tutor and listen to a few basic objectives. Then, they must complete some very basic objectives that teach some of the basic character abilities. Much of the tutorial dialogue may be skipped, but the skills necessary to continue must be demonstrated. In this way, the game teaches new skills but never requires tedious instruction. The player learns primarily by doing. All of this is done in the shadow of a very visible path onto the rest of the game so the user never loses sight of where they need to go.

In-game interfaces. In-game interfaces are used primarily to deliver necessary status feedback and to perform less-frequent functions. We measure effectiveness with more traditional lab usability testing techniques and desirability with attitude measurements such as surveys (see next section for example).

Some PC games make extensive use of in-game interfaces to control the game. For example, simulation and RTS games can be controlled by keyboard and mouse presses on interface elements in the game. Usability improvements in these interfaces can broaden the audience for a game by making controls more intuitive and reducing tedious aspects of managing the game play. In-game tutorial feedback can make the difference between confusion and quick progression in learning the basic mechanisms for playing. In this situation, iterative usability evaluations become a key methodology for identifying problems and testing their effectiveness (see next section for example).

Many complex PC and console video games make frequent use of in-game feedback and heads-up displays (HUD) to display unit capabilities and status. For example, most flight combat games provide vital feedback about weapons systems and navigation via in-game displays. Without this feedback, it can be difficult to determine distance and progress towards objectives, unit health and attack success. This feedback is crucial for player learning and satisfaction with the game. With increasing game complexity and 3-D movement capabilities, these displays have become a crucial part of many game genres. Usability testing is required to establish whether users can detect and correctly identify these feedback systems (see Pagulayan, Steury, Fulton, et al., 2003 for detailed example of usability testing the HUD in *MechWarrior 4: Vengeance,* 2000).

Mapping input devices to functions. A learnable mapping of buttons, keys, or other input mechanisms to functions is crucial for enjoying games. We measure effectiveness with usability techniques and desirability with attitude measurements. Without learnable and intuitive controls, the user will make frequent mistakes translating their desires into onscreen actions. We have seen consistently that these kinds of mistakes are enor-

mously frustrating to users, because learning to communicate one's desires through an eight-button input device is not very fun. The selection of keys, buttons, and other input mechanisms to activate particular features is often called "control mapping." Players tend to feel that learning the control mapping is the most basic part of learning the game. It is a stepping-stone to getting to the fun tasks of avoiding obstacles, developing strategies, and blowing things up.

By contrast to other ease of use issues, evaluating the control mapping may involve as much subjective measurement as behavioral observation. Button presses are fast, frequent, and hard to collect automatically in many circumstances. Furthermore, problems with control mappings may not manifest themselves as visible impediments to progress, performance, or task time. Instead, they may directly influence perceptions of enjoyment, control, confidence, or comfort. Due to differences in experience levels and preferences between participants, there may also be significant variation in attitudes about how to map the controls.

Dissatisfaction with the controller design can also be a central factor that limits enjoyment of all games on a system. For example, the results of one set of studies on the games for a particular console game system were heavily influenced by complaints about the system's controller. Grasping the controller firmly was difficult because users' fingers were bunched up and wrists were angled uncomfortably during game play. Ratings of the overall quality of the games were heavily influenced by the controller rather than the quality of the game itself.

Because of these concerns and the importance of optimizing control mappings, we recommend testing control mappings with both usability and attitude-assessment methodologies.

Challenge

Challenge is distinct from ease of use and is measured almost exclusively with attitude-assessment methodologies. This can be a critical factor to the enjoyment of a game, and obviously can be highly individualized and is rightly considered subjective.

Consumers may have difficulties distinguishing the appropriate kinds of challenge that result from calculated level and obstacle design, from the difficulty that is imposed by inscrutable interface elements or poor communication of objectives. In either case, the result is the same. If not designed properly, the player's experience will be poor. Thus, it is up to the user-testing professional to make measurement instruments that evaluate the appropriateness of the challenge level independent of usability concerns. In one example, Pagulayan, Steury, Fulton, et al. (2003) used attitude-assessment methodologies to determine the final design of the career mode in *RalliSport Challenge* (2002). In this situation, finding a solution to the design problem was not necessarily related to ease of use issues, or other usability-related issues. The final design was based on what was most fun and appropriately challenging for users.

Pace

We define pace as the rate at which players experience new challenges and novel game details. We measure this with attitude-measurement methodologies.

Most designers will recognize that appropriate pacing is required to maintain appropriate levels of challenge and tension throughout the game. You might think of this as the sequence of obstacles and rewards that are presented from the start of the game to the end. However, the way a designer will address pace will depend on a variety of issues, including game type, game genre, and their particular vision for the gameplay experience. In a tennis game, pace can be affected by a number of things, including the number of cut-scenes in between each point, to the actual player and ball movement speed (Pagulayan & Steury, 2004).

One group at Microsoft uses a critical juncture analogy to describe pacing. As a metaphor, they suggest that the designer must attend to keeping the user's attention at 10 seconds, 10 minutes, 10 hours, and 100 hours. The player can always put down the game and play another one, so one must think creatively about giving the user a great experience at these critical junctures. Some games excel at certain points but not others. For example, the massively multiplayer game may have the user's rapt attention at 10 seconds and 10 minutes. The fact that hundreds of thousands pay $10 per month to continue playing indicates that these games are very rewarding at the 100-hour mark. Anyone who has played one of these games can tell you that they are extremely difficult and not too fun to play at the 10-hour mark. At this point, you are still being killed repeatedly. That is no fun at all. Arcade games obviously take this approach very seriously. Though they may not scale to 100 hours, good arcade games attract you to drop a quarter and keep you playing for long enough to make you want to spend another quarter to continue.

Pacing may also be expressed as a set of interwoven objectives much like the subplots of a movie. Again, *Banjo Kazooie* (2000) provides an excellent example of good pacing. Each level in *Banjo Kazooie* contains the tools necessary to complete the major objectives. Finding the tools is an important part of the game. New abilities, objectives, skills, and insights are gradually introduced as the player matures. While progressing towards the ultimate goal (of vanquishing the evil witch and saving the protagonist's sister), the player learns to collect environmental objects that enable them to fly, shoot, become invincible, change shape, gain stamina, add extra lives, and unlock new levels. This interlocking set of objectives keeps the game interesting and rewarding. Even if one is unable to achieve a particular goal, there are always sets of sub goals to work on. Some of which may provide cues about how to achieve the major goal.

Summary

Attitude methodologies are better apt to measure factors such as overall fun, graphics, sound, challenge, and pace. The typical iterative usability is an exploratory exercise designed to uncover problem areas where the designers intentions do not match the users expectations, as a result, we typically choose not to use usability test to assess fun or challenge. When attempting to assess attitudinal issues as overall fun and challenge, we make use of a survey technique that affords testing larger samples. Internally, we have adopted the term *Playtest* or sometimes *Consumer Playtest* for this technique. At the

same time, we use typical iterative usability methods to determine design elements that contribute to or detract from the experience of fun.

USER RESEARCH IN GAMES

Introduction to Methods—Principles in Practice

In the following section, we propose various methodologies and techniques that attempt to accurately measure and improve games game usability and enjoyment. Many of the examples are taken from techniques used and developed by the Games User Research group at Microsoft Game Studios.

Our testing methods can be organized by the type of data being measured. At the most basic level, we categorize our data into two types: behavioral and attitudinal. Behavioral refers to observable data based on performance, or particular actions performed by a participant that one can measure. This is very similar to typical measures taken in usability tests (e.g., time it takes to complete a task, number of attempts it takes to successfully complete a task, task completion, etc.). Attitudinal refers to data that represent participant opinions or views, such as subjective ratings from questionnaires or surveys. These are often used to quantify user experiences. Selection of a particular method will depend on what variables are being measured and what questions need to be answered.

Another distinction that is typically made is between formative and summative evaluations, which we apply to our testing methods as well. Formative refers to testing done on our own products in development. Summative evaluations are benchmark evaluations, done either on our own products or on competitor products. It can be a useful tool for defining metrics or measurable attributes in planning usability tests (Nielsen, 1993), or to evaluate strengths and weaknesses in competitor products for later comparison (Dumas & Redish, 1999).

While these methods are useful, they do not allow us to address issues with extended gameplay, that is, issues that may arise after playing the game for a couple of days or more. This is problematic, because one of the key challenges in game design is longevity. With the competition, the shelf life of a game becomes very limited.

Usability techniques. Traditional usability techniques can be used to address a portion of the variables identified as important for game design. In addition to measuring performance, we use many standard usability techniques to answer how and why process-oriented questions. For example, how do users perform an attack, or why are controls so difficult to learn? We use (a) structured usability tests, (b) rapid iterative testing and evaluation (RITE; Medlock, Wixon, Terrano, & Romero, 2002), and (c) other variations and techniques, including open-ended usability tasks, paper prototypes, and gameplay heuristics. For clarity of presentation, each technique will be discussed separately, followed by a case study. Each case study will only contain information pertinent to a specific technique, thus examples may be taken from a larger usability test.

Structured usability test. A structured usability test maintains all the characteristics that Dumas & Redish (1999) proposed as common to all usability tests: (a) goal is to improve usability of the product, (b) participants represent real users, (c) participants do real tasks, (d) participant behavior and verbal comments are observed and recorded, and (e) data are analyzed, problems are diagnosed, and changes are recommended. We have found that issues relating to expectancies, efficiency, and performance interaction are well suited for this type of testing. Some common areas of focus for structured usability testing are in game shell screens, or control schemes. The game shell can be defined as the interface where a gamer can determine and or modify particular elements of the game. This may include main menus and options screens (e.g., audio, graphics, controllers, etc.).

An example that uses this method is in the *MechCommander 2* (2001) usability test. Portions of the method and content have been omitted.

Case Study: MechCommander 2 Usability Test. MechCommander 2 (MC2) is a PC RTS game where the gamer takes control of a unit of *mechs* (e.g., large giant mechanical robots). One area of focus for this test was on the *'Mech Lab*, a game shell screen where mechs can be customized (see Fig. 37.2). The gamer is able to modify weaponry, armor, and other similar features and is limited by constraints such as heat, money, and available slots.

The first step in approaching this test was to define the higher-order goals. Overall, the game shell screens had to be easy to navigate, understand, and manipulate, not only for those familiar with mechs and the mech universe, but also for RTS gamers who are not familiar with the mech universe. Our goal was for gamers to be able to modify and to customize mechs in the *'Mech Lab*.

As mentioned, one of the most important steps in this procedure is defining the participant profiles. Getting the appropriate users for testing is vital to the success and validation of the data since games are subject to much scrutiny and criticism from its gamers. To reiterate, playing games is a choice. For

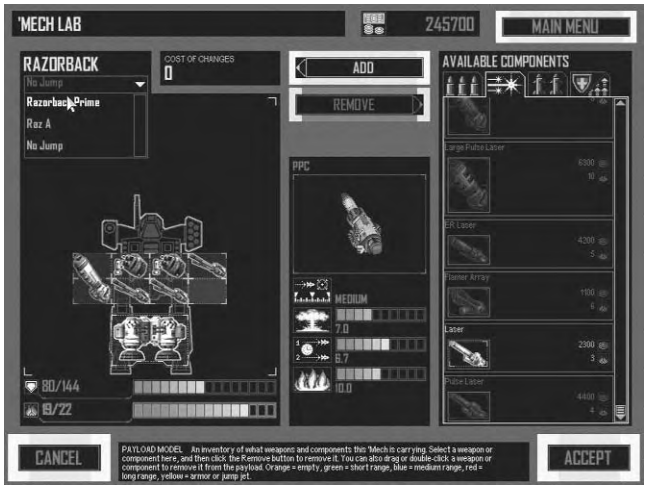


FIGURE 37.2. Screenshot of the 'Mech Lab in MechCommander 2.

MechCommander 2 (2001), we defined two participant profiles which represented all of the variables we wanted to cover. The characteristics of interest included those who were familiar with RTS games (experienced gamers) and those who were not RTS gamers (novice gamers). We also wanted gamers that were familiar with the mech genre, or the mech universe. Overall, we needed a landscape of gamers that had some connection or interest that would make them a potential consumer for this title, whether through RTS experience, or mech knowledge.

Tasks and task scenarios were created to simulate situations that a gamer may encounter when playing the game. Most importantly, tasks were created in order to address the predefined higher order goals. Participants were instructed to talk aloud, and performance metrics were recorded (e.g., task completion, time). The following are examples from the usability task list.

1. Give the **SHOOTIST** jumping capabilities.

This task allowed us to analyze participant expectations. To succeed in this task, one had to select a "CHANGE WEAPONS" button from a different game shell screen, which brought them into the *'Mech Lab*. If the task was to change a weapon on a mech, the terminology would probably have been fine. Thus, this task had uncovered two main issues: (a) could they get to the *'Mech Lab* where you modify the mech, and (b) were they able to discover the process of modifying the mech. It was accurately predicted that gamers would have difficulties with the button terminology. Thus, that button was changed to "MODIFY MECH."

To change the components (e.g., add the jump jets), participants could either select the item and drag it off the mech or select the item and press the "REMOVE" button (see Fig. 38.2). One unexpected issue that arose was that participants unknowingly removed items because the distance required for removing an item was too small. The critical boundary that was implemented was too strict. In addition, participants had difficulties adding items by dragging and dropping because the distance required for adding an item was too large (e.g., the item would not stay on the mech unless it was placed exactly on top of the appropriate location). Appropriate recommendations were made and implemented.

2. Replace the **MG Array** with the **Flamer**.

One of the constraints presented for modifying a mech was heat limit. Each weapon had a particular heat rating. For example, if the heat limit for a mech is 35 and the current heat rating is 32 only weapons with a rating of 3 or fewer could be added. In this task, the "Flamer" had a heat rating much larger than the "MG Array," thus making impossible to accomplish this task without removing more items. The issues here were the usability of the heat indicator, heat icons, and the discoverability of heat limit concept. None of the participants figured this out. Recommendations included changing the functionality of the Heat Limit Meter and adding better visual cues to weapons that exceed the heat limit. Both of these changes were implemented.

Rapid iterative testing and evaluation method (RITE). Medlock et al. (2002) documented another common usability method used by the Games User Research Group at Microsoft Game Studios, which they refer to as the RITE method. In this method, fewer participants are used before implementing changes, but more cycles of iteration are performed. With RITE, it is possible to run almost two to three times the total sample size of a standard usability test. However, only one to three participants are used per iteration with changes to the prototype immediately implemented before the next iteration (or group of one to three participants).

The goal of the RITE method is to address as many issues and fixes as possible in a short amount of time in hopes of improving the gamer's experience and satisfaction with the product. However, the utility of this method is entirely dependent on achieving a combination of factors (Medlock et al., 2002). The situation must include (a) a working prototype; (b) the identification of critical success behaviors, important, but not vital behaviors, and less important behaviors; (c) commitment from the development team to attend tests and immediately review results; (d) time and commitment from development team to implement changes before next round; and (e) the ability to schedule and run new participants as soon as the product has been iterated. Aside from these unique requirements, planning the usability test is very similar to more traditional structured usability tests.

It is very helpful to categorize potential usability issues into four categories: (a) clear solution, quick implementation, (b) clear solution, slow implementation, (c) no clear solution, and (d) minor issues. Each category has implications for how to address each issue. In the first category, fixes should be implemented immediately and should be ready for the next iteration of testing. In the second category, fixes should be started in hopes that it can be tested by later rounds of testing. For the third and fourth category, more data should be collected.

The advantage of using the RITE method is that is allows for immediate evaluation and feedback of recommended fixes that were implemented. Changes are agreed upon and made directly to the product. If done correctly, the RITE method affords more fixes in a shorter period. In addition, by running multiple iterations over time, we are potentially able to watch the number of usability issues decrease. It provides a nice, easily understandable, and accessible measure. In general, more iterations of testing are the better. However, this method is not without its disadvantages. In this situation, we lose the ability to uncover unmet user needs or work practices, we are unable to develop a deep understanding of gamer behaviors, and we are unable to produce a thorough understanding of user behavior in the context of a given system (Medlock et al., 2002).

The following example demonstrates how the RITE method was used in designing the *Age of Empires II: The Age of Kings* (1999) tutorial. Again, portions of the method and content have been omitted (see Medlock et al., (2002) for more details).

Case Study: Age of Empires II: The Age of Kings Tutorial. *Age of Empires II: The Age of Kings* (*AoE2*; 1999) is an RTS game for the PC in which the gamer takes control of a civilization spanning over a thousand years, from the Dark Ages through the late medieval period. In this case study, a working prototype of the tutorial was available, and critical concepts and behaviors were defined. Also, the development team was committed to attending each of the sessions. They were committed to quickly implementing agreed upon changes. Finally, the resources for scheduling were available. The key element in this situation for

TABLE 37.2. Age of Empires Example Concepts and Behaviors Categorized into Three Concepts Using the RITE Method (Medlock, Wixon, Terrano, & Romero, 2002)

Essential Concepts/Behaviors	Important Concepts/Behaviors	Concepts/Behaviors of Lesser Interest
• movement	• queuing up units	• using hotkeys
• multiselection of units	• setting gathering points	• using mini-map modes
• "fog of war"	• garrisoning units	• using trading
• scrolling main screen via mouse	• upgrading units through technology	• understanding sound effects

success was the commitment from the development team to work in conjunction with us.

In the *AoE2* tutorial, there were four main sections: (a) marching and fighting (movement, actions, unit selection, the fog of war[1]), (b) feeding the army (resources, how to gather, where to find), (c) training the troops (use of mini-map, advancing through ages, build and repair buildings, relationship between housing and population, unit creation logic), and (d) research and technology (upgrading through technologies, queuing units, advancing through ages). Each of these sections dealt with particular skills necessary for playing the game. In essence, the tutorial had the full task list built in. In the previous case study, this was not the case.

At a more abstract level, the goals of the tutorial had to be collectively defined (with the development team). In more concrete terms, specific behaviors and concepts that a gamer should be able to perform after using the tutorial were identified, then categorized into the three levels of importance: (a) essential behaviors that users must be able to perform without exception, (b) behaviors that are important, but not vital to product success, and (c) behaviors that were of lesser interest. Table 37.2 lists some examples of concepts and behaviors from each of the three categories. This is an important step because it indirectly sets up a structure for decision rules to be used when deciding what issues should be addressed immediately, and what issues can wait.

The general procedure for each participant was similar to other usability tests we often perform. If participants did not go to the tutorial on their own, they were instructed to do so by the specialist.

During the session, errors and failures and were recorded. In this situation, an error was defined as anything that caused confusion. A failure was a considered an obstacle that prevented participants from being able to continue. After each session, a discussion ensued among the specialist and the development team to determine what issues (if any) warranted an immediate change at that time.

In order to do this successfully, certain things had to be considered. For example, how can one gauge how serious an issue is? In typical usability tests, the proportion of participants experiencing the error is a way to estimate its severity. Since changes are made rapidly here, the criteria must change to the intuitively estimated likelihood that users will continue to experience the error. Another thing to consider is clarity of the issue, which was assessed by determining if there is a clear solution. We have often found that if issues do not have an obvious solution then the problem not fully understood. Finally, what errors or failures were essential, important, or of lesser interest. Efforts of the development team should be focused on issues related to the essential category when possible.

At this point, we broke down the issues into three groups. The first group included issues with a solution that could be quickly implemented. Every issue in this group was quickly implemented before the next participant was run. The second group consisted of issues with a solution that could not be quickly implemented. The development team began working on these in hopes it could be implemented for later iterations of the test. Finally, there were issues with no clear solutions. These issues were left untouched because more data was needed to assess the problem at a deeper level (e.g., more participants needed to be run). Any fixes implemented in the builds were kept as each participant was brought in. Thus, it was possible that many of the participants experienced a different version of the tutorial over the duration of testing.

Overall, seven different iterations were used across 16 participants. Fig. 37.3 represents the number of errors and failures recorded over time. The number of errors and failures gradually decreased across participants as new iterations of the build were introduced. By the seventh and final iteration of the build, the errors and failures reliably were reduced to zero.

Although we feel that the *AoE2* tutorial was an enormous success, largely due to the utilization of the RITE method, the method does have its disadvantages and should be used with caution. Making changes when issues and solutions are unclear may result in not solving the problem at all, while creating newer usability problems in the interface. We experienced this phenomena a couple of times in the *AoE2* study.

Also, making too many changes at once may introduce too many sources of variability and create new problems for users. Deducing specifically the source of the new problem becomes very difficult. A related issue is not following up changes with enough participants to assess whether or not the solution really addressed the problem. Without this follow up, there is little evidence supporting that the implementations made were appropriate (which is a problem with traditional usability methods as well). The last thing to consider is that other important usability issues that may surface less frequently are likely to be missed. Using such small samples between iterations allows for the possibility that those less occurring issues may not be detected.

Variations on usability methods. Now that we have presented two general types of usability testing, it is worthy to mention some variations on these methods: (a) open-ended tasks, (b) paper prototyping, and (c) gameplay heuristics.

[1]The *fog of war* refers to the black covering on a mini-map or radar that has not been explored yet by the gamer. The fog of war lifts once that area has been explored. Use of the fog of war is most commonly seen in RTS games.

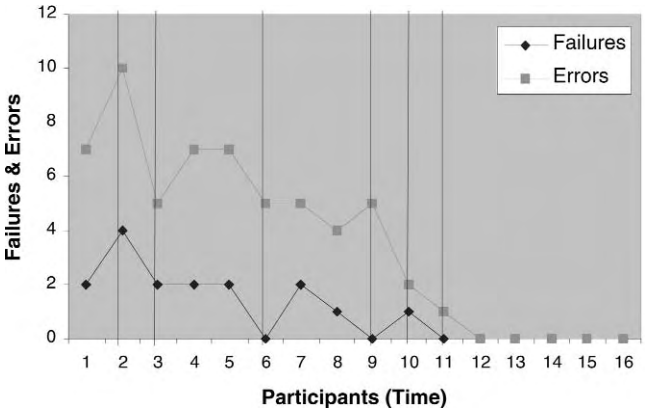


FIGURE 37.3. A record of failures and errors over time for the Age of Empires Tutorial when using the RITE method (Medlock et al., 2002).

In general, it is often recommended that tasks in usability tests be small, with a specified outcome (e.g., Nielsen, 1993). However, we have found situations where the inclusion of an open-ended task yields important data as well. In many usability studies, we often include an open-ended task where participants are not instructed to perform or achieve anything in particular. In other words, there is no specified outcome to the participant. These tasks can be used to analyze how gamers prioritize certain tasks or goals, in a nonlinear environment. These tasks are also useful in situations where structured tasks may confound the participant experience or situations where we are interested in elements of discovery. An example of an open-ended task is as follows.

1. Play the game as if you were at home. The moderator will tell you when to stop.

This example was taken from a usability test on *Halo: Combat Evolved* (2001; Pagulayan, Steury, Fulton et al., 2003). Participants were presented with a mission with no instruction other than playing as if they were at home. Traditional usability metrics were not tracked or used. Instead, the focus was watching players and the tactics and strategies they employed while playing through the game. Results demonstrated that novice players would start firing at enemies as soon as they were visible, which was not how the designers intended combat to occur. The design intent was for combat to occur at closer ranges.

By allowing participants to play through the mission with no structured task, designers were able to detect the strategies players would employ. As a result, several changes were made to the gameplay to encourage players to engage in much closer combat (see Pagulayan, Steury, Fulton et al., 2003 for more details).

Prototyping, heuristic evaluations, and empirical guideline documents are other techniques we often use when more time-consuming testing cannot be done. In practice, these techniques do not differ when used on games. Nielsen (1993) categorized prototyping and heuristic evaluations as "discount usability engineering," and we would agree. We also tend to view empirical guideline documents in a similar manner. Empirical guideline documents are essentially lists of usability principles for particular content areas based on our collective experience doing user-testing research. Examples of some of these content areas include console game shell design, PC game shell design, PC tutorial design principles, movement, aiming, and camera issues in first and third person shooter games, and online multiplayer interfaces. Desurvire, Caplan, and Toth (2004) developed a list of heuristics targeted at computer and video games. These have been broken down into four general categories: game play, game story, game mechanics, and game usability. For the full list of gameplay heuristics, see Desurvire et al., (2004).

Survey Techniques

The use of surveys has been explored in great depth (e.g., Bradburn & Sudman, 1988; Couper, 2000; Labaw, 1981; Payne, 1979; Root & Draper, 1983; Sudman, Bradburn, & Schwarz, 1996) and is considered a valid approach for creating an attitudinal data set as long as you ask questions that users are truly capable of answering (see Root & Draper, 1983). We conduct surveys in the lab and over the Internet in order to collect attitudinal data regarding gameplay experiences, self-reported play behaviors, preferences, styles, and motivations.

Formative Playtest surveys. We combine hands-on play with surveys to create a formative lab method called "Playtest." In nearly all instances, questioning follows the pattern of a forced choice question (sometimes a set of forced-choice questions) followed by a more open-ended question encouraging the participants to state the reasons behind their response. The goal of a Playtest is to obtain specific information about how consumers perceive critical aspects of a game and provide actionable feedback to game designers (Davis et al., 2005). The basic Playtest method is used formatively to compare a product at time one and time two during its development. While a modified version of the basic method is used cumulatively to compare a summative evaluation to other, relevant games.

Playtests have several advantages and characteristics that differentiate them from usability studies and online survey research. For one, they focus mainly on perceptions. A larger sample size is required for statistical power and generalization of the results to the target audience. In most cases, 25 to 35 participants are used as a pragmatic trade-off between confidence and resource constraints. Specialized studies may require more. Statistical representations of the attitudinal measures are used to identify potential strengths and weaknesses that need to further be explored, provide information about the extent or severity of an attitude or to describe trends of interest that come to light.

Conducting the studies in the lab as opposed to online, allow us to test in-progress games during early development. It also allows us to take more control of the testing situations and monitor a limited amount of play behaviors that can be paired with the attitudinal data.

On the other hand, there are a few notable disadvantages. First, structured lab studies limit the amount of gameplay that can be tested in a single study. An artificial lab scenario or time line may prevent the engineer from collecting critical information about the game experience. This is a classic validity trade-off present in most lab studies. Second, the amount of prepara-

tion that is required to design a Playtest questionnaire takes some time. The study plan cannot be adjusted during the course of the study. The engineer must be intimately aware of the development team's questions to include all of the relevant questions in the survey. This requires domain knowledge in games and experience in survey design.

Second, the data can sometimes be difficult to interpret if the problem is not major or if the game facet is novel. Sometimes the data does not provide specific causes for the problem, making it difficult to nail down the best solution. Follow-up studies, (Playtest or usability) are often recommended (or necessary) to parse out these issues. Finally, the approachable nature of the data is ripe for misinterpretation by team members who may not be familiar with the current state of the game, design details, or the methods used in Playtest. It has been rightfully asserted that usability tests do not necessarily require a formal report upon completion (Nielsen, 1993). We believe that may be less true for using this kind of technique. Careful and consistent presentation formats are required. Instructions and contextual information that will help with interpretation should be included, along with some form of qualitative data (e.g., open-ended comments). Metrics from games in development should focus on problem detection and often cannot be used for predictive purposes.

Development of the Playtest surveys. Several steps have been taken to increase the effectiveness and efficiency of Playtest. First, core questions for each genre, play style, and core game feature have been developed, using a multistep iteration and evaluation process. This process included construct development, survey item development, formative evaluation with subject matter experts, and formative evaluation of question validity using Cognitive Interviewing techniques and statistical validation. While many survey items have been validated, most Playtests requires new items to be created in order to meet the specific needs of that particular game and study goal. Therefore, a repository of previously used customized questions has been created, along with guidelines for writing new survey items.

Type of Formative Studies

There are essentially three types of formative studies that are conducted during development: (a) the critical facet test, (b) the initial experience test, and (c) the extended Playtest. We also run tests that do not easily fit into any of these categories, including subtle variations on the above, a few cases of large sample observationally based studies, and studies that focus on games from a more conceptual level.

As in the usability section, for clarity of presentation, each technique will be discussed separately, followed by a case study. Each case study will only contain information pertinent to a specific technique, thus examples may be taken from a larger Playtest.

Critical facet Playtest. Games often take the form of repeating a core experience within an array of different context and constraints. A driving game is always about aiming an object that is hurtling through space. The critical facet Playtest focuses directly on that core experience and making sure that it is fun. While the core experience can often be assessed in usability testing, Playtesting is necessary to assess attitudes and perceptions about the core experience.

The following example demonstrates how a critical facet test was applied to the critical facets of *Oddworld: Munch's Oddysee* (2001), an Xbox game. *Oddworld: Munch's Oddysee* is a platform/adventure game that allows you to switch back and forth between two main characters as they proceed through the increasingly difficult dangers of Oddworld on a quest to save Munch's species from extinction. The core gameplay is exploring the realm by running, jumping, and swimming through the environment. In this case, there were concerns about the user's visual perspective—which we typically call the camera. Does the camera support exploration of the environment?

Case study: Munch's Oddysee, Camera. Previous usability testing with the *Oddworld: Munch's Oddysee* (2001) determined that while some users indicated dissatisfaction with the behavior of the camera, other participants chose not to mention it at all while engaged in the open-ended usability tasks. The camera's behavior was programmed to create maximal cinematic effects (e.g., sometimes zooming out to show the size of an area) and attempt to enhance gameplay. The camera would often show a specific view with the intent of showing you what was behind the next door or on the other side of the wall while still keeping the main character in view. While many users liked the look, style, and behavior of the camera, users often wanted more control over the behavior of the camera. Indeed some participants would actively say things such as, "That looks really cool right now [after the camera had done something visually interesting] but I want the camera back pointing this way now." Because feedback from the usability lab contained both positive and negative feedback, the development team did not see the usability data as conclusive. Further, changing the camera would be a major cost to the game in terms of redesign and redevelopment time.

After having played the game for an hour, 25 participants were asked for general perceptions of the game. More specific questions followed. Questions related to the camera were asked in the latter portion of the questionnaire because previous experience in the usability lab had shown that merely mentioning the camera as part of a task would often cause participants previously silent on the subject to vociferously criticize aspects of the camera's behavior. With the knowledge that we wanted to factor out any priming-related effects, two analyses were conducted.

The first analysis was based on the response to the questions related to the behavior of the camera itself. Nearly half of the participants (46%) indicated that the camera did not give them enough flexibility of control.

The second analysis went back through individual responses to determine the attitudes of those participants who mentioned the camera before the survey first broached the subject. Forty-three percent of the participants were found to have mentioned the camera in a negative fashion prior to being asked specifically about the camera questions.

Based on this data and other anecdotal evidence, the development team chose to give the players more flexibility of camera control. The result was more frequent use of a camera behavior

we termed a *third-person follow camera.* The behavior of this camera had the double advantage of being more easily controlled by users and of conforming to a set of behaviors more often expected by users. It maintained focus on the main character without major adjustments to point of view (e.g., to "look over a wall" or "behind a door"). Other camera behaviors (e.g., still camera behaviors that pan with the character) are still a part of the game but have been localized to areas where these alternative camera behaviors can only create an advantage for the user.

Initial experience Playtest. As with many things, first impressions are a key component of overall satisfaction with a game. Given that many games are a linear experience/narrative, there is a lot of value in obtaining attitudinal data related to the first portions of gameplay. Lessons learned at first are often applied throughout the game. Obviously, the later portions of the game will never be experienced unless the first portions of the game are enjoyed.

The following example explains how a set of formative initial experience tests were run for *MechCommander 2*, the RTS game described earlier in the chapter. The earlier usability test focused on the choices that users could make prior to starting a mission, whereas this test focused on in-game components, such as user's ability to take control of their squad and lead them through battles, while experiencing satisfaction and motivation to continue playing.

Case study: MechCommander 2, Initial Missions. Twenty-five participants, who were representative of the target market, were brought onsite and were asked to begin playing from the first mission of the game. After each mission, participants were asked to stop playing and report their impressions of the fun, the excitement, and the clarity of objectives. Participants also indicated their comfort level with the basics of controlling the game. The participants were able to play through as many as three missions before the session ended. For purposes of brevity, this case study will focus on the results related to the first mission.

Although the first mission had been designed to offer tutorial elements in a brief and fun way, the experience was generally not satisfying to participants. Approximately a third of the participants had a poor initial impression of the game. They felt that the challenge was too easy and that the mission was not exciting. Furthermore, there were some problems related to clarity of goals. By combining the survey results with opportunistic observation, it was noted that some participants moved their units into an area where they were not intended to go. This disrupted their experience and limited their ability to proceed quickly. Responses to more open-ended questions indicated that some of the core gameplay components and unique features did not come across to users. Some users complained about the standard (predictable or commonplace) nature of the game. Finally, several participants complained about the fact that they were being taught everything and wanted to turn the tutorial off.

A number of actions were selected from team insight and user testing recommendations. First, the team decided to separate the tutorial missions from the required course of the game—in order to save experienced players from the requirement of completing the tutorial. Second, the scope of the first mission was expanded so that users would have more time in their initial experience with the game. Third, the clarity of objectives was improved via minor interface changes. Fourth, addressing the same issue, the design of the map on which the mission took place was revamped to require a linear approach to first mission that limited the likelihood of users becoming lost. Fifth, the amount and challenge of combat was increased. Finally, one of the unique components of the game was introduced. The new mission included the ability to call in support from off-map facilities. Despite all these changes, the mission was still targeted to be completed within approximately 10–15 minutes.

A follow-up Playtest was intended to verify the efficacy of the changes. Twenty-five new participants were included in the study. Results from this test indicated that the design changes had created a number of payoffs. Far fewer participants felt the mission was too easy (13% vs. 33%), only 3% indicated that the mission was not exciting, measures of clarity of objectives improved, and surprisingly, there was no drop off on ratings of comfort with basic controls because of the tutorial aspects being improved. Results were not a total success, as some participants were now rating the mission as too hard while others in response to open-ended questions complained about the overly linear nature of the mission (e.g., there was only one way to proceed and few interesting decision related to how to proceed through the mission). In response to these results, the development team decided to retune the first mission to make it a little easier, but not to address the comments related to the linearity of the mission. The second mission of the game was less linear, and so it was hoped that, with the major constraints to enjoyment removed, most participants would proceed to the second mission and find the mission goals to be more engaging. The data from the second mission was rated far more fun than the first mission, validating their assumption.

Extended Playtests. While the initial experience with a game is very important, the success or failure of a game often depends on the entire experience. In an attempt to provide game developers with user-centered feedback beyond the first hour of gameplay, we conduct extended Playtests that test several hours of consecutive gameplay. This is done by conducting Playtest studies that run for more than two hours, or by having participants participate in more than one study. Attitudinal data is taken at significant time or experience intervals (e.g., after a mission). Participants can also provide self-directed feedback about a specific point in the game at anytime during the session. Basic behaviors and progress (e.g., completed level 4 at 12:51 p.m.) can be recorded by Playtest moderators and used in conjunction with the attitudinal data.

Focus groups. In addition to usability and survey techniques, we sometimes employ other user-centered design methods that are qualitative in nature. Focus groups provide an additional source of user-centered feedback for members of the project team. Our Playtest facilities offer us the opportunity to precede a candid user-discussion with hands-on play or game demonstration. In this setting, participants can elaborate on their experience with the product, provide feedback about the user-testing process, speculate about the finished version of the

product, generate imaginative feature or content ideas, or find group consensus on issues in a way that models certain forms of real-world shared judgments.

For example, a focus group discussion regarding a sports game revealed that users had varying opinions about what it meant for a game to be a simulation versus an arcade-style game. Additionally, they had many different ideas as to what game features made up each style of game. From this conversation, we learned a few of the Playtest survey questions were not reliable and needed revision.

Besides the general limitations inherent in all focus group studies (Greenbaum, 1988; Krueger, 1994), the focus groups described here typically include practical limitations that deviate from validated focus group methods. Generally, focus group studies comprise of a series of 4 to 12 focus groups (Greenbaum, 1988; Krueger, 1994); however, we typically run one to three focus groups per topic. This makes the focus group more of a generative research tool rather than a confident method for identifying real trends and topics in the target population. Fewer group interviews make the study more vulnerable to idiosyncrasies of individual groups.

A *Combined Approach*

As you can see, we have provided a number of different techniques that we use in game development and evaluation. The presentation has been structured in such way to clearly demonstrate the differences between the techniques we use. The important thing to realize is that no one method exists independent of other methods. The following is a brief case study on an Xbox game called *Blood Wake* (2001) that demonstrates how some of these techniques can be used together.

Case Study: Blood Wake. Blood Wake (2001) is a vehicular combat console game using boats and a combination of powerful weapons. Our initial main concern was the aiming mechanism and the controls—users must be able to control their boat and destroy the enemy. A secondary concern was related to the initial missions, goals, gameplay, and general look and feel of the game

The first problem we addressed was the aiming mechanism. Users were presented with two different types of aiming schemes: auto-aim and manual-stick aim. In the auto-aim scheme, the target reticule automatically snapped to an enemy that was within a reasonable distance allowing the player to maneuver their boat while shooting the enemy. In the manual-stick aim scheme, the user had to manually aim the reticule using the right stick on the gamepad with no assistance. In previous usability studies, we have found difficulties with the manual aiming mode. However, this method is used quite often in popular games, thus the general belief was that it was better and preferred. Many felt the auto-aim method would be too easy.

We decided to run a within-subjects design Playtest to assess which scheme users liked better. Our results showed that 73% of participants liked the auto-aim, whereas only 38% liked the manual-stick aim. This was proof enough to convince the developers to focus their efforts on the auto-aim method.

Although we solved the aiming controversy, unoptimized navigation controls made it difficult to assess the fun and challenge of the game. Poor attitudinal ratings could be explained by the gameplay or the poorly tuned boat controls. In order to optimize these controls, we headed to the usability labs. There we presented participants with two different control schemes. In one scheme, throttle and steering were both mapped to the left analog stick (push forward/back for throttle; push left/right for steering). This enabled the user to drive and steer with one finger. In another scheme, the throttle was mapped to the right trigger, and steering was on the left analog stick. Separating the steering from the throttle would map more directly onto familiar mental models for driving and steering cars. Previous data indicated that neither scheme was clearly superior, so the goal of this test was to optimize the controls for both schemes.

Participants were asked to perform common maneuvers in the game. The following are examples taken from the usability task list:

1. Once the game starts, you should see a bunch of boxes that form a path in the distance. As fast as you can, try to follow the trail of items and collect as many boxes as you can without turning around if you get off course.
2. Somewhere on this map, there is a straight line of boxes. Find the straight line of boxes then try to slalom (like a skier) through the line of the boxes. Try to do this as fast as you can without collecting any of the boxes.

As you can see from the two examples, the tasks were very performance based. Participants were instructed to think aloud while we measured task performance and task time.

By the end of the study, we were able to provide detailed information to the team about how to optimize the controls. Some of the issues that were addressed included the sensitivity of the steering, controlling boat speed, and difficulties moving the boat in reverse. Iteration on the controls was followed by multiple Playtests that focused on user perceptions of five of the initial missions. The results of the Playtest validated our earlier recommendations and allowed us to tweak the controls even more.

By running multiple Playtests, we were also able to get data and feedback on gameplay. After each mission, we elicited feedback on the perceived difficulty, clarity of objectives, attitudes about the environment, and general fun ratings. This allowed us to pinpoint where participants were running into problems, such as not knowing where to go next and not understanding how to complete objectives. This feedback led to dramatic changes in the design of the missions.

CONCLUSION

The need for the continued development of user-centered design methods in video games has indeed arrived. Games drive new technologies and affect an enormous amount of people. In addition, games represent a rich space for research areas involving technology, communication, attention, perceptual-motor skills, social behaviors, virtual environments, just to name a few. It is our position that video games will eventually develop an intellectual and critical discipline, like films, which would result in continually evolving theories and methodologies of game de-

sign. The result will be an increasing influence on interface design and evaluation. This relationship between theories of game design and traditional HCI evaluation methods has yet to be defined, but definitely yields an exciting future.

As mentioned earlier, user-centered design methods are beginning to find their way into the video games industry. Commercial game companies, such as Ubisoft Entertainment, Electronic Arts, Inc., and Microsoft Corporation, employ some level of user-centered methodologies to their game development processes, and this presumably will grow. However, this is not to say that video games are qualitatively different from other computer applications. There are just as many similarities as differences to other computer fields that have already benefit from current user-centered design methods. Thus, it makes sense to utilize these methods when applicable, but also to adapt methods to the unique requirements that we have identified in video games.

In this chapter, we emphasized the difference between games and productivity applications in order to illustrate the similarities and differences between these two types of software applications. We also chose to reference many different video games in hopes that these examples would resonate with a number of different readers. Case studies were included to try to demonstrate in practice how we tackle some of the issues and challenges mentioned earlier in the chapter. It is our intention that practitioners in industry, as well as researchers in academia, should be able to take portions of this chapter, and adapt them to their particular needs when appropriate, similar to what we have done in creating the actual methods mentioned in this chapter. That said, we are upfront that most, if not all of our user-testing methods are not completely novel. Our user-testing methods have been structured and refined based on a combination of our applied industry experience, backgrounds in experimental research, and of course, a passion for video games. This allows us to elicit and utilize the types of information needed for one simple goal and to make the best video games that we can for as many people as possible.

ACKNOWLEDGMENTS

We would like to thank the Games User Research Group at Microsoft Game Studios. In addition, we would like to express our gratitude to Paolo Malabuyo, Michael Medlock, and Bill Fulton for their insights and input on the creation of this chapter, and Kathleen Farrell and Rally Pagulayan for their editing assistance and reviewing early drafts. The views and opinions contained in this chapter are those of the authors and do not necessarily represent any official views of Microsoft Corporation.

References

Age of Empires II: Age of Kings [Computer software]. (1999). Redmond, WA: Microsoft Corporation.

Animal Crossing [Computer software]. (2002). Redmond, WA: Nintendo of America.

Banjo Kazooie [Computer software]. (2000). Redmond, WA: Nintendo of America.

Bradburn, N. M., & Sudman, S. (1988). *Polls and surveys: Understanding what they tell us*. San Francisco: Jossey-Bass.

Blood Wake [Computer software]. (2001). Redmond, WA: Microsoft Corporation.

Brute Force [Computer software]. (2003). Redmond, WA: Microsoft Corporation.

Cassell, J., & Jenkins, H. (2000). Chess for girls? Feminism and computer games. In J. Cassell & H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and computer games* (pp. 2–45). Cambridge, MA: The MIT Press.

Chaika, M. (1996). *Computer games marketing bias*. ACM Crossroads. Retrieved date, from http://www.acm.org/crossroads/xrds3-2/girlgame.html

Combat Flight Simulator [Computer software]. (1998). Redmond, WA: Microsoft Corporation.

Couper, M. P. (2000). Web surveys: A review of issues and approaches. *Public Opinion Quarterly, 64*, 464–494.

Crawford, C. (1982). *The art of computer game design*. Berkeley, CA: Osborne/McGraw-Hill.

Csikszentmihalyi, M. (1990). *Flow—The Psychology of Optimal Experience*. New York, NY: Harper & Rowe.

Davis, J. P., Steury, K., & Pagulayan, R. J. (2005). A survey method for assessing perceptions of a game: The consumer playtest in game design. *Game Studies: The International Journal of Computer Game Research, 5*(1),. Retrieved date, from http://www.gamestudies.org/0501/davis_steury_pagulayan/

Desurvire, H., Caplan, M., & Toth, J. (2004). Using heuristics to improve the playability of games. *CHI 2004: Conference on Human Factors in Computing Systems,* Vienna Austria. ACM's Special Interest Group on Computer-Human Interaction.

Diablo [Computer software]. (2000). Paris, France: Vivendi Universal.

Diddy Kong Racing [Computer software]. (1997). Redmond, WA: Nintendo of America.

Dumas, J. S., & Redish, J. C. (1999). *A practical guide to usability testing (Rev. ed.)*. Portland, OR: Intellect Books.

Entertainment Software Association. (2005). 2005 Sales, demographics and usage data: Essential facts about the computer and video game industry. Washington, DC: Entertainment Software Association.

Final Fantasy X [Computer software]. (2001). Tokyo, Japan: Square Co., Ltd.

Flight Simulator 2000 [Computer software]. (2000). Redmond, WA: Microsoft Corporation.

Funge, J. (2000). Cognitive modeling for games and animation. *Communications of the ACM, 43*(7), 40–48.

Greenbaum, T. L. (1988). *The practical handbook and guide to focus group research*. Lexington, MA: DC Heath and Company.

Halo: Combat Evolved [Computer software]. (2001). Redmond, WA: Microsoft Corporation.

Halo 2 [Computer software]. (2004). Redmond, WA: Microsoft Corporation.

Harvest Moon [Computer software]. (2003). Redmond, WA: Nintendo of America.

Hecker, C. (2000). Physics in computer games. *Communications of the ACM, 43*(7), 35–39.

Herz, J. C. (1997). *Joystick nation: How videogames ate our quarters, won our hearts, and rewired our minds*. New York, NY: Little, Brown and Company.

Hyman, P. (2005). State of the industry: Mobile games. *Game Developer Magazine, 12* (14), 11–16.

IDG Entertainment, (2005, August). *IDG Entertainment casual games market report*. Oakland, CA: IDG Entertainment.

International Game Developers Association. (2005). *IDGA 2005 Mobile games whitepaper.* Paper presented at the Game Developers Conference 2005 by the Mobile Games SIG, San Francisco, CA.

Interactive Digital Software Association (2000). State of the industry report (2000–2001). Washington, DC: Interactive Digital Software Association.

Internet Backgammon, Windows XP [Computer software]. (2001). Redmond, WA: Microsoft Corporation.

Jablonsky, S., & DeVries, D. (1972). Operant conditioning principles extrapolated to the theory of management. *Organizational Behavior and Human Performance, 7*, 340–358.

Kent, S. L. (2000). *The first quarter: A 25-year history of video games*. Bothell, WA: BWD Press.

Kroll, K. (2000). Games we play: The new and the old. *Linux Journal, 73es*. Retrieved date, from http://www.acm.org/pubs/articles/journals/linux/2000-2000-73es/a26-kroll/a26-kroll.html

Krueger, R. A. (1994). *Focus groups: A practical guide for applied research*. Thousand Oaks, Ca: Sage.

Labaw, P. (1981). *Advanced questionnaire design*. Cambridge, MA: Abt Books Inc.

Laird, J. E. (2001). It knows what you're going to do: Adding anticipation to a Quakebot. *Proceedings of the Fifth International Conference on Autonomous Agents, Canada*, 385–392.

Lepper, M., Greene, D., & Nisbett, R. (1973). Undermining children's intrinsic interest with extrinsic rewards. *Journal of Personality and Social Psychology, 28*, 129–137.

Lepper, M. R., & Malone, T. W. (1987). Intrinsic motivation and instructional effectiveness in computer-based education. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, Learning and Instruction III: Conative and Affective Process Analyses*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Levine, K. (2001). *New opportunities for Storytelling*. Paper presented at the Electronic Entertainment Exposition, Los Angeles, CA.

Madden NFL '06 [Computer software]. (2005). Redwood City, CA: Electronic Arts Inc.

Malone, T. W. (1981). Towards a Theory of Intrinsic Motivation. *Cognitive Science, 4*, 333–369.

MechCommander 2 [Computer software]. (2001). Redmond, WA: Microsoft Corporation.

MechWarrior 4: Vengeance [Computer software]. (2000). Redmond, WA: Microsoft Corporation.

Medlock, M. C., Wixon, D., Terrano, M., & Romero, R. (2002, July). *Using the RITE Method to improve products: a definition and a case study.* Orlando, FL: Usability Professionals Association.

Microsoft Word 2003 [Computer software]. (2005). Redmond, WA: Microsoft Corporation.

Moore, K., & Rutter, J. (2004). Understanding Consumers' Understanding of Mobile Entertainment. In K. Moore & J. Rutter (Eds.), *Proceedings of Mobile Entertainment: User-centred Perspectives* (pp. 49–65). *CRIC*, University of Manchester.

MTV Music Generator [Computer software]. (1999). Warwickshire, United Kingdom: The Codemasters Software Company Limited.

NFL2K [Computer Software]. (1999). San Francisco, CA: Sega of America, Inc.

Nielsen, J. (1993). *Usability engineering*. San Francisco, CA: Morgan Kaufmann.

Oddworld: Munch's Odyssey [Computer software]. (2001). Redmond, WA: Microsoft Corporation.

Pagulayan, R., Gunn, D., & Romero, R. (2006). A Gameplay-Centered Design Framework for Human Factors in Games. In W. Karwowski (Ed.) *2nd Edition of International Encyclopedia of Ergonomics and Human Factors* (pp. 1314–1319). London: Taylor & Francis.

Pagulayan, R. J., & Steury, K. (2004). Beyond usability in games. *Interactions, 11*(5), 70–71.

Pagulayan, R. J., Steury, K., Fulton, B., & Romero, R.L. (2003). Designing for fun: User-testing case studies. In M. Blythe, A. Monk, K. Overbeeke, & P. Wright (Eds.), *Funology: From usability to enjoyment* (pp. 137–150). Netherlands: Kluwer Academic Publishers.

Payne, S. L. (1979). *The art of asking questions*. Princeton, NJ: Princeton University Press.

Pokemon Crystal [Computer software]. (2000). Tokyo, Japan: Nintendo Japan.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Reading, MA: Addison-Wesley.

Provenzo, E. F., Jr. (1991). *Video kids: Making sense of Nintendo*. Cambridge, MA: Harvard University Press.

RalliSport Challenge [Computer Software]. (2003). Redmond, WA: Microsoft Corporation.

Root, R. W., & Draper, S. (1983, December 12–15). *Questionnaires as a software evaluation tool.* Paper presented at the ACM CHI, Boston, MA.

Sidel, P. H., & Mayhew, G. E. (2003). The Emergence of Context: A Survey of MobileNet User Behaviour. Retrieved date, from http://www.mocobe.com/pdf/EmergenceofContext1.pdf

Steers, R. M., & Porter, L. W. (1991). *Motivation and work behavior* (5th ed.). New York, NY: McGraw-Hill, Inc.

Sudman, S., Bradburn, N. M., & Schwarz, N. (1996). *Thinking about answers: The application of cognitive processes to survey methodology*. San Francisco, CA: Jossey-Bass.

The Legend of Zelda: The Wind Waker [Computer Software]. (2003). Redmond, WA: Nintendo of America.

Williams, J. H. (1987). *Psychology of women: Behavior in a biosocial context (3rd ed.)*. New York: W. W. Norton.

Yuen, M. (2005, September). The Tipping Point: The convergence of wireless and console/PC game design. Game Developers Magazine.

Part

V

DESIGNING FOR DIVERSITY

·38·

THE DIGITAL DIVIDE: THE ROLE OF GENDER IN HUMAN COMPUTER INTERACTION

Joel Cooper and Matthew B. *Kugler*
Princeton University

Introduction **764**
The Digital Divide **764**
What the Digital Divide Is Not 765
Understanding the Roots of the Digital Divide **765**
In the Beginning 766
A Designer's Bias 767
Greater, Anxiety, and Poorer Performance 768
The Hazards of Low Expectations 769
Mere Awareness: Stereotype Threat 770
Compounding the Problem: Gender Differences in Attribution 770
Dismantling the Divide **772**
Software Design Can Limit the Gender Stereotype . . . 772
Adjusting the Social Context 773
Changing the Conclusions 773
References **773**

INTRODUCTION

This is the age of the computer. Whether we use a personal computer to balance our checkbook, operate a mainframe that sorts signal intercepts for our government, or work the cash register at the local Burger King, the steady hum of computer technology permeates our daily existence. Comparison shopping can now be done in seconds online with a few mouse clicks, rather than in days of real-world driving. University bookstores now compete not only with local equivalents, but also with online merchants on other continents often with favorable consequences to the wallet and purse. The cost of distributing ideas has reached an all-time low, forever altering political dialogue, and websites have brought together far-flung communities dedicated to everything from Japanese cartoons to satanic cults. According to the latest available figures from the U.S. Census, 61.8% of households in the United States have home computers and 54.7% have Internet access (U.S. Census, 2005).

This prevalence masks a persistent problem, however. While the 61.8% figure is impressive from the perspective of merely a decade ago, it is nonetheless true that the increase in computers has not been uniform across every subgroup in society nor has it affected all groups in the same way. To the contrary, the computer revolution has left some groups behind. A person with a bachelor's degree is 30% more likely to own a computer than a person with only a high-school education. A household with an income of $75,000–$99,999 has a 90% chance of owning a computer; one with an income of $25,000–$49,999 has a 67% chance. There are also racial differences in computer ownership. White and Asian Americans are over 20% more likely to own a computer than Black and Hispanic Americans (U.S. Census, 2005). Moreover, in the last decade of the 20th century, the gap in computer ownership between African Americans and Whites widened. These differences persist even when controlling for income. It has been shown that owning a computer leads to dramatic advantages on academic test scores. It is particularly interesting that, controlling for the number of computers in a particular household, wealthy Americans and White Americans gained even more of an advantage than poor and minority students (Atwell & Battle, 1999).

A divide also exists between men and women, with women not enjoying the benefits of the technological revolution on par with men (Cooper & Weaver, 2003). The difficulties women face while using computers are sweeping. They are underrepresented in their use and ownership of computers (Pinkard, 2005; Wilson, Wallin, & Reiser, 2003; Yelland & Lloyd, 2001), take fewer technology classes in high school and college (Pinkard, 2005), are far less likely to graduate college with degrees in IT fields and, most significantly, enjoy interacting with computers much less than do men (Mitra, Lenzmeir, Steffensmeir, Avon, Qu, & Hazen, 2000).

Computers are becoming central to more jobs every year. Current estimates suggested that by 2010, 25% of all new jobs in the public and private sectors will be technologically oriented (AAUW, 2000). However, even more important, computers play a role in all of the basic activities of life from banking, to shopping, to—increasingly—voting. Decades ago, computer innovation was driven by the space program, the cold war, and military technology. Now, a new car's computer technology is more than 1,000 times more powerful than what guided the Apollo moon missions (Alliance of Automobile Manufacturers, 2006). Computers are inescapable. With all of this in mind, it is a societal problem that the path to computer efficacy is more difficult for the poor, ethnic minorities, and women (Wilson, Wallin, & Reiser, 2003).

THE DIGITAL DIVIDE

Discrimination against women, at least in certain domains, has deep and complex roots. Understanding the basis of such discrimination is important and complex. The roots of the digital divide share some commonalities with discrimination that women have faced in employment and professional advancement but also have their own distinct origins. The use of computers in the home, classroom, and workplace is only a few decades old, which affords us the opportunity to gain a glimpse at the genesis of the particular problem of the gender divide in information technology.

In the late 1970s, computers began to replace television as the technological innovation in the classroom. By the 1980s they were ubiquitous in education and on their way to becoming a fixture in most households. In this context, Wilder, Mackie, and Cooper (1985) surveyed school children to assess their attitudes toward computers. They found a large difference in the degree to which boys and girls were attracted to the computer. As early as kindergarten, boys indicated more positive attitudes about computer technology than girls. These small attitudinal differences became dramatic in the fifth grade and continued to grow through the middle- and high-school years (Wilder, Mackie, & Cooper, 1985). Computer use had just begun to spread into the mainstream of public life, and numerous explanations for the difference were considered. Wilder et al. hoped that the gender differences in regards to computers were either an artifact of the particular geographic area studied in the investigation or something that would diminish as technology became more widely accessible. It was easy to hope that the problem would fix itself in those days; public education is a great equalizer.

This was not to be the case. Disturbing effects discovered in the 1980s persisted into the 1990s. The clearest data was not on the question of usage, but on anxiety. In a host of domains, both young girls and older women reported that computers are not creators of fun and amusement but rather the source of apprehension. Weil, Rosen and Sears (1987) reported that about 1 in 3 adults in the United States experienced what they called "computerphobia"—adverse anxiety reactions to the use of computers. Dembrot and her colleagues were among the first to investigate the imbalance in computer anxiety as a function of gender. They found that female college students expressed considerably more anxiety about computers than did their male counterparts (Dambrot, Watkins-Malek, Silling, Marshall, & Garver, 1985; see also Temple & Lips, 1989). This finding was replicated frequently throughout the 1990s (i.e., Colley, Gale, & Harris, 1994; Todman & Dick, 1993). In the late 1990s, these differences between males and females were as ubiquitous as they were in the 1980s, with females from elementary school

grades to university graduates expressing greater anxiety and negative attitudes (Brosnan, 1998; Whitley, 1997).

Now, in the first decade of the 21st century, there are some promising signs that the gender gap in computer technology may be weakening. The U.S. Census showed marked increase in computer use by women, especially in the use of the Internet and e-mail and in the workplace. Nonetheless, despite the increased use, women continue to lag behind men in feelings of competence with the computer. They also continue to suffer greater anxiety about using information technology and have fewer positive attitudes about working and playing with the computer than do men (Colley & Comber, 2003; Schumacher & Morahan-Martin, 2001). Surveying school-age children and comparing their responses to those collected more than a decade ago, Colley and Comber (2003) found that girls' interest in computer applications improved, but that girls continue to like the computer less than boys do. When given a chance to use computers in the voluntary world outside of school, girls use the computers less frequently than do boys. Similarly, Mucherah (2003) recently reported that teenage girls feel far less involved with computers and enjoy them less than boys of comparable ages. At Princeton University, researchers asked incoming college students about their reactions to computers (Cooper & Weaver, 2003). Despite having a highly capable and academically accomplished sample, they found that the young women were far less confident of their ability with computers than were the young men. The incoming female undergraduates reported feeling significantly less comfortable with computers than the men did, even though most of them had taken computer classes in their high schools and more than 80% of them had taken higher-level mathematics, including calculus. That any differences were seen in such circumstances is very discouraging, and the effects were not small.

Those same researchers also asked incoming students to imagine that they were going to take a course in psychological statistics. They presented the following question: Suppose that you were asked to complete a statistics homework assignment on the computer. How comfortable would you feel in doing that assignment? These highly capable students again differed based on gender. Men felt that they would be comfortable completing the assignment while women felt uncomfortable (Cooper & Weaver, 2003). Therefore, the lack of confidence just noted is not merely an abstract concept. Even in the context of a specific example, women were just not as sure of their abilities as men were. We can easily imagine that the difference might have been even greater had 4 out of 5 of the women not already completed courses in calculus.

The digital divide is a worldwide problem. Much of the previous research was conducted in the United States. Other studies from Western Europe and other highly developed countries show similar effects. Data have been gathered in Great Britain (Colley, Gale, & Harris, 1994), Australia (Okebukola & Woda, 1993), Canada (Temple & Lips, 1989), and Spain (Farina, Arce, Sobral, & Carames 1991), always with the same result. In a review of this literature for the International Association for the Evaluation of Educational Achievement, Reinen and Plomp (1997) concluded that, "concern about gender equity is right. . . . Females know less about information technology, enjoy using the computer less than male students and perceive more problems with . . . activities carried out with computers in schools" (p. 65).

As interest in this issue has intensified, additional international data have led to the same conclusion. Recent data reported from Romania (Dundell & Haag, 2002), Egypt (Abdelhamid, 2002) and Italy (Favio & Antonietti, 2002), for example, continue to show the persistence of the digital divide in a wide array of educational systems around the globe. Although there are some exceptions, (i.e., Solvberg, 2002, in Norway) gender differences have been remarkably durable.

What the Digital Divide Is Not

The gender gap is less about total hours using a computer than about using a computer voluntarily for enjoyment and comfort with information technology. In schools, it is not about total number hours spent in front of the computer screen, but rather the interference of computer anxiety with the ability and excitement to learn. In the workplace, the digital divide is not about the magnitude of use, but rather about women's reactions to the technology with which they interact. It is about their comfort, attitudes, and levels of anxiety. Women use computers at their jobs more than men do. The use of the computer as typewriter and cash register, for example, necessarily requires human computer interaction (HCI) in the workplace, and with women holding far more service and administrative support jobs than men, their computer use is relatively high. In fact, in 2003, 63% of women used computers for their jobs; whereas only 51% of men did so.

Exposure in the workplace and in the school has not ended the disparity between men and women in terms of their levels of comfort using a computer, attitudes about computers, and willingness to use computers in contexts in which computer use is not required. Especially in educational settings, anxiety with using computers can not only result in a feeling of discomfort, but also can lead to less-adequate performance with the computer and the material that was supposed to be learned more enjoyably and efficaciously with computer technology.

UNDERSTANDING THE ROOTS OF THE DIGITAL DIVIDE

The digital divide is not caused by lack of use, nor is it due to differences in economic status, social class, or heredity. We also assume that differences based on biological sex play, at most, a negligible role in accounting for the differences. Rather, we see the different reactions to information technology to be rooted in the socialization of boys and girls as they learn to cope with the social constructions that form the norms, rules, and expectations for their gender. As a heuristic guide to understanding the digital divide, we propose a model that described a series of factors whose result is differential attitudes and differential comfort levels with the use of computers in contemporary society.

The model, which we will describe in more detail in the following sections, takes as its starting point the idea that there exists in our social world entrenched stereotypes of the behaviors and attitudes that are appropriate for children and adults

of each gender. Boys are supposed to be more eager to play with computers than girls. The most important consequence of this stereotype is that girls will experience more anxiety when playing with, or learning from, computers; thus making it difficult for them to have pleasant and successful computer interactions. This will happen whether or not girls accept the stereotype as valid. Girls who accept the stereotype as valid will be harmed by what is referred to as the "self-fulfilling prophecy" (Merton, 1948; Rosenthal & Jacobson, 1968). Ironically, girls who do not believe that the stereotype is true will nonetheless experience anxiety with computers because of the phenomenon known as "stereotype threat" (Steele & Aronson, 1995). The mere knowledge that the stereotype exists and other members of society believe it sets in motion processes that lead to confirmation of the stereotype. As the model showed, a girl who knows that there is a stereotype predicting poor computer competency on her part will experience more computer anxiety and, in the end, poorer performance and more negative attitudes about computers. This, in turn, will lead to anxiety and a greater chance of failure.

Our model also shows that different attributional patterns for boys and girls contribute to the cycle that perpetuates the digital divide. Because of the different interpretations that boys and girls are taught with regard to success in achievement domains, the stereotype about the relation of gender to computer use may become reinforced and more resistant to change. As Fig. 38.1 suggests, the dilemma is a self-reinforcing cycle in which boys, typically to their advantage, and girls, to their disadvantage, become enveloped in the veil of the gender stereotype for computing.

In the Beginning

Undeniably, gender stereotypes abound. Like most stereotypes, they were created by society over an extended time, and even though they are now undesirable, they are reluctant to be dismantled. Regardless of whether or not they are true, stereotypes have dramatic impact on behavior. For example, in most western societies, we share common societal expectations about the toys boys and girls are supposed to play with. We do not expect to see the war characters in our favorite toy store sharing shelf space with dress-up dolls and doll strollers. To the contrary, we expect that the warriors will be near the cars, trains, and space heroes. The dolls will be near the carriages, play houses, and play schools. Boys will find their toys in the former section; girls in the latter. In reality, of course, there is gender overlap such that some boys find their favorite toy in the section with dolls and carriages. And many girls are in toy-heaven when confronted by the cars and trucks. However, in general, there is a strong effect for gender stereotyping of toys, based on and reinforced by, adult expectations of what is expected to be interesting and pleasing to boys and girls.

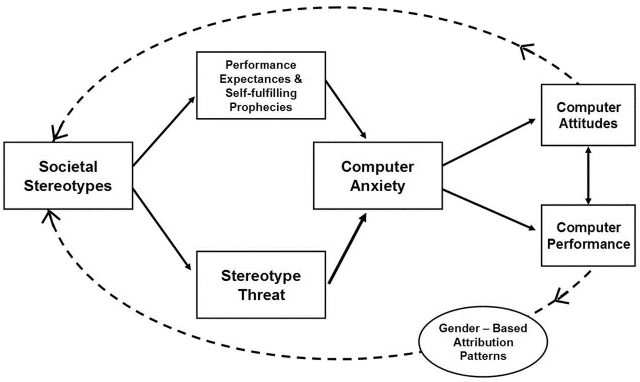


FIGURE 38.1. A logical model of digital divide.

Gender stereotypes abound in the classroom as well. Regardless of whether or not the stereotypes are true, we see mathematics, science, and technology as the province of boys more than girls. Girls write well and are interested in literature and poetry. Computers are the bedrock of information technology and, as we have seen from surveys of children and adults described above, in established democracies and developing nations, we have a similar stereotype about who enjoys and benefits from computers. It is not immediately apparent why gender stereotypes developed for computers become nearly identical to stereotypes about science and mathematics. Although the algorithms that comprise computer software are complex and mathematically sophisticated, and computers burst into our consciousness in large-scale space and science ventures, most computer users do not interact with computers at that level. A screen, keyboard and mouse pad form the basis of the interfaces that most people have with computers. Why did the use of computers become associated with gender?

The answer to that question is multidetermined and a full analysis is beyond the scope of the current chapter. However, we can isolate one of the causes of the gender stereotype in the introduction of the computer into the educational system. The classroom is a ubiquitous melting pot and its influence on children's attitudes is profound. When educators first looked to computers to supplement their normal educational methods, they made an understandable, though fundamental, error. They drew their inspiration from the world of the video game and video arcade. The best examples of popular computer games in the 1980s were not the increasingly rich variety currently available, but the far less diverse sampling of the video arcade and the early Nintendo and Sega gaming systems. While these programs drew their contexts from a multiplicity of domains, everything from medieval combat to futuristic space adventures, what most had in common was an emphasis on competitive responding. As games grew more elaborate, story lines were increasingly incorporated to keep children's interest in space adventures, sports, and battles.

Educators have always searched for ways to make learning more efficient and more enjoyable. That computers can give students an interactive experience makes them an obvious and attractive addition to the classroom. Computer-software manufacturers turned out hundreds of programs designed to assist teachers in delivering instruction in every discipline from art to zoology. They most likely contemplated how to design such programs and wondered what children wanted. One thing that was obvious at this time was that childred would rush to finish—or ignore entirely—their homework for a chance to hit the arcade. Video-game designers were posting large profits and the growth

of their industry seemed to be limited only by rapidly disappearing technical issues. If children would voluntarily spend hours navigating the story line of *Kings Quest*, it seemed fair to assume that instruction delivered in the same game-like format would be popular.

What went unnoticed was predominantly boys visited arcades and spent hours playing their favorite video games. Turning classrooms into video arcades by adopting software that resembles video games might have been attractive to most young boys, but it presented a problem for girls. Every benefit that was gained among video-game players by making learning software the image of video games was a deficit to those who did not relate to or enjoy such games.

One of the earliest analyses of the issue of computer technology in the classroom was that of Lepper and Malone (1987). They asked girls and boys what they liked about computers. The responses of boys matched intuitions. Boys liked activities that were in the form of games. Story-lines featuring sports, war, and space were popular. Boys liked eye-hand coordination and competition. They preferred their feedback in the form of flashing lights, blaring sounds, and explosions. Girls were different. They disliked competitive programs, the videogame emphasis on reflexes, and the dramatic end games. What they wanted was better learning tools. Frequent and clear feedback, preferably not in the form of explosions, was also favored. Girls appreciated computers as learning tools, but found the game focus very disheartening. With computer-assisted instruction (CAI) programs like *Word Invasion, Demolition Division,* and *Slam Dunk Math*, it is clear that boys were more precisely targeted than girls. Chappel (1996) reported that in the real worlds of education and business, programs overwhelmingly favor male identification and male interests. Yet such programs have been one of the major ways in which children are introduced to computers. The grand attempt to "make learning fun" has been premised on an unfortunate, gender-biased definition of the word "fun" that is not widely applicable.

A Designer's Bias

Why is there such a strange disconnect between program designers and female users? (See Lynn, Raphael, Olefsky, & Bachen, 2003 for a discussion.) There are many possibilities. Men are more interested in computer programming than women, and one could say that male programmers are more likely to produce male-focused software, but that is both uncharitable and hard to test. Of far greater interest is the possibility that software designers implicitly assume that their users will be male and tailor their work accordingly. In everyday communication, we often adapt our tone to our audience. If we think our audience is hostile, we are likely to act in a more hostile manner ourselves (Snyder & Swann, 1978). Similarly, if a male college student believes that the female he is speaking to is attractive, he is likely to act in a friendly and inviting manner (Snyder, Tanke, & Berscheid, 1977). In HCI, a software designer is interacting with a communication partner, albeit a strange one. In the mind's eye of the programmer is an eventual user. Someone is sitting at the front end of the computer screen, answering questions, and interacting with the program. But who is that user? It is possible that the software designer uses stereotypical beliefs to portray the most likely consumer for whom he or she is preparing an educational or entertainment product. In this view, computer programmers write software as though they are communicating with boys because their automatic representation of the gender of the user is male.

Huff and Cooper (1987) investigated the possibility that the gender of the typical user influences the communication process and, accordingly, the characteristics of the software that is produced. They asked teachers in the New Jersey public schools to design software to help seventh-grade children learn the appropriate use of commas. The teachers were given one of three different instructions. One group of teachers was asked to design the software for seventh-grade boys, another for seventh-grade girls, and the third, most interestingly of all, were just told to tailor their work for seventh-grade "students." These teachers were all well versed in the likes and dislikes of seventh graders. Their designs were always fascinating and the two gender-specific conditions produced the kind of program concepts that Lepper and Malone (1987) would have expected. In describing her boy-directed idea, one teacher wrote:

> here is an opportunity to enjoy the world, do sports, and learn English grammar at the same time. Your child will enjoy shooting cannons and competing for the highest score. After playing with this program, you child will use commas in a natural and correct manner.

The group writing for girls had no difficulty in guessing the right features to include. A typical response was expressed by one teacher when she described her program as, "Two girls go on a shopping trip to a record shop to find music for a dance being given at school. They converse with each other and make decisions about what to buy. The use of commas and rules involved are taught through this trip. Reinforcement is available in worksheet form." The activity is social, not military, and lacks the boyish embellishments that girls find anxiety inducing.

The teachers recognized that they needed to write vastly different programs to motivate students of different genders. While their results appear to rely heavily on gender stereotypes, they do recognize the problem and produce programs that are appropriate to the point of caricature. Based solely on these conditions, one would expect that teachers would be able to find a happy medium when working for "students."

The "student" condition, however, is precisely where the problem arises. Teachers designing without a specified gender did *not* assume that half their users would be male and half would be female. Their answers were resoundingly like the programs that had been written for boys and nothing like the programs that had been written when girls were the focus of attention. A typical description in this condition began, "Here's a fast-paced program for your arcade-game lovers. Just what the teenager spends his quarters on! . . . . Sentences zip across the screen—some correctly punctuated with commas, some not. Correct sentences are "zapped" off the screen by your students as they try to be on the roster of top scorers."

Programs that most teachers wrote for "students" were nearly identical to programs that other teachers wrote for boys. All of the programs were coded and assessed by independent raters and then subjected to a multidimensional scaling analysis.

The results showed that the programs written for students were statistically indistinguishable from the programs written for boys on a dimension that ranged from "learning tool" to "toy." Both were markedly on the "toy" end of the dimension. Programs written when girls were the focus of attention were written as learning tools and were significantly different from both the boy and student programs (Fig. 38.2).

Greater, Anxiety, and Poorer Performance

Even though it was clear that most educational programs were designed with the boy definition of fun in mind, it still remained to be seen what effect these programs had on girls compared to more gender-neutral—though far less common—alternatives. Cooper, Hall, and Huff (1990) examined this issue using *Demolition Division,* a program intended to teach division in a stereotypically boy manner by employing war-related imagery, competition, and eye-hand coordination. Its manufacturer described it as "an opportunity to practice the division of problems [*sic*] in a war game format. Tanks move across the screen as guns from bulkheads are fired by the students as he answers the problem. Hits and misses (correct and incorrect answers) are recorded at the bottom of the screen." The researchers had middle-school boys and girls learn division with either this program or another one, *Arithmetic Classroom,* a CAI program that lacked all of the features of a stereotypically boy-focused game but taught essentially the same information. The students worked with the program for several minutes in a computer cluster in their school. Following the exercise, the children filled out a questionnaire assessing their liking for the CAI-learning program as well as their level of anxiety and stress. When they returned to the classroom, their ability to perform division problems was assessed.

The data showed that girls liked the *Demolition Division* program less often than boys did, and the girls also were considerably more anxious at the conclusion of the CAI lesson. In addition, the level of anxiety was negatively correlated with performance: the more anxious the student, the less she had learned. In *Arithmetic Classroom*, however, the results were quite different. Remember that this program had no competitive elements, no eye-hand coordination tasks, and no war-story plot line. The anxiety levels of girls using this program were not any higher than those of boys. Girls felt quite comfortable and experienced slightly less anxiety than did the boys. The results are shown in Fig. 38.3. When the computer software had the formal elements that boys enjoy, girls showed the typical pattern of the digital divide: They experienced stress and anxiety and, consequently, did not perform as well. There was no evidence for the gender divide when the program had the formal features that girls enjoy.

Nearly a decade after Cooper et al., another group of researchers found a similar effect in a more elaborate computer game. In this case, the variable of concern was not flashing lights or explosions, but identification. Littleton, Light, Joiner, Messer, and Barnes (1992) worried that many of the storyline CAI games did not include characters that girls could relate to. These researchers targeted a popular CAI game called *King and Crown*, which taught a series of spatial reasoning skills in an adventure format. The characters in the game, however, were primarily warriors and the game was aggressive. Boys learned the skills necessary for the game and fully succeeded in the adventure approximately 50% of the time. Girls were successful only 8% of the time. Littleton et al. found that the male-oriented world of *King and Crown* caused the girls to disidentify, become anxious, and withdraw. Yet, the problem was with the context, not the content. When performance was examined in another program, one that taught the same skills in a gender-neutral context, girls and boys performed equally well. Now, 50% of both genders completed the game. Taken together, the Cooper et al. (1990) and Littleton et al. studies suggested that girls are not innately inferior at learning through computer programs. The problem is in the design of the game. Programs rich in boy-favorable elements are not efficient learning tools for girls; their anxiety level increases while their interest and performance decrease. Also important is that boys do not do worse on the programs that allow girls to do better.

The problem is not one confined to young children in the lower grades. The problem may start there, but it persists into adulthood. Surveys have revealed negative attitudes and higher

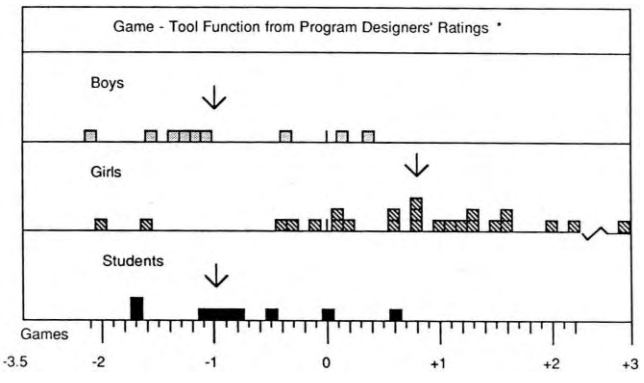


FIGURE 38.2. Level of computer anxiety after learning from a male oriented (Demolition Division) and Control (Arithmetic Classroom) Computer Assisted Learning program. (Source: Huff & Cooper).

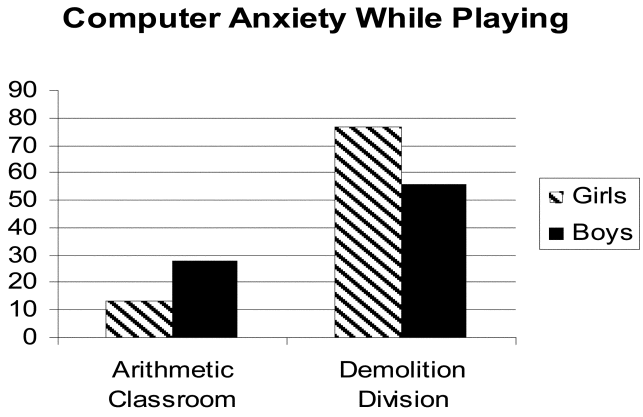


FIGURE 38.3. Results of multi-dimensional scaling depicting programs written by teachers for instructing boys, girls and students. The arrows depict the central tendency of each condition. (Source: Huff & Cooper).

computer anxiety among females in college, the workplace, and among retirees (Zhan, 2005). Experimental data collected with college students also support the negative effects of male-favored software on women's levels of computer anxiety. For example, in a study by Robinson-Staveley and Cooper (1990), men and women students at Princeton University played the game of *Zork,* in which players compete to find a buried treasure in an adventure-game format. Women reported a high degree of stress while playing the game and, in turn, performed poorly. Male students, on the other hand, performed considerably better and did not experience computer anxiety.

Another interesting finding in the Robinson-Staveley and Cooper (1990) study is that the social context of computing has a substantial effect on the experience of computer anxiety. In their research, gender differences in performance occurred only in the presence of other people. If the students were asked to solve the *Zork* adventure in complete privacy, the women did well (better than the men) and experienced only a little computer anxiety. Similarly, middle-school girls in the *Demolition Division* study did not experience more computer anxiety than boys if they worked with the CAI program without the presence of others (Cooper et al., 1990).

The social context also matters when the gender composition of learning groups is considered. Girls learning in the presence of boys suffer from increased computer anxiety and learn less. Light, Littleton, Bale, Joiner, and Messer (2000) had boys and girls work with a mildly competitive problem-solving game in which the players' task was to reach a geographical location without being captured by monsters. The children worked in groups of two, either same sex or opposite sex dyads. Light et al. (2000) found that, overall, boys performed better than girls in this game. However, in same-sex dyads, the difference in performance was small. In mixed-sex dyads, the difference was enhanced. Boys' performance was markedly improved relative to their performance in the same-sex group while girls' performance showed significant decrements.

Nicholson, Gelpi, Young, and Sulzby (1998) examined the interactions that occurred when first-grade girls and boys were asked to work together on a computer task. They found that, in mixed-gender groups, girls were likely to have their competence and/or their work criticized or laughed at compared to girls working with other girls. It is unsurprising that taking an already somewhat anxious person and putting her in a position where she may be subject to ridicule or competition worsens performance, but it is still an important consideration. While the centrality of social context in these effects relates more to usage environment than program design, these findings underscore the socially constructed nature of the gender gap. In a more friendly and accepting environment, girls can prosper.

What we have seen through the analysis of patterns in education software is just part of one path through the model we propose. Gender-stereotypes have led to the creation of programs that increase the girls' computer anxiety. This anxiety then lowers their computer performance, making them more skeptical about their computer ability and reaffirming the stereotype. However, software and peer environment are not the only means through which society can affect a woman's computer success. What are the effects of an authority figure, a parent, a teacher, a boss, holding such a stereotype?

The Hazards of Low Expectations

Reactions to stereotypes can vary. First, the stereotype can be believed. There is good reason to believe that gender-based stereotypes can have the power of the self-fulfilling prophecy, creating further evidence for the stereotype. The classic self-fulfilling prophecy study was conducted in a classroom setting by Rosenthal and Jacobson (1968). These researchers convinced teachers that a new test of intelligence could predict which of their students were likely to experience sudden improvements in their academic ability during the next school year. Some students were identified as likely to show these "spurts," but for the majority, nothing was said. When they tested the students at the end of the year, they found that students that the teachers had expected to show considerable improvement did perform much better on standardized tests than students not labeled that way, especially among younger students. The interesting part of this study was that the test Rosenthal and Jacobson administered was entirely bogus and the results were intentionally randomized. The teachers responded differently to students whom they expected to perform well. While no data shows exactly the nature of this difference, imagination provides a host of possible explanations; the teachers could have been more encouraging, more attentive, and more supportive of those they "knew" could do well. This study has long been seen as a warning to teachers to be careful in labeling their students either positively or negatively. Students are attuned to the expectations of their superiors and can be persuaded to either try harder or give up based on what is expected.

Word, Zanna, and Cooper (1974) showed that stereotypes about groups of people also impact peoples' performance. Their study made use of implicit racial stereotypes about African Americans. In the context of a job interview, Word et al. showed that the negative racial stereotypes that White job interviewers held about the traits and capabilities of Black candidates subtly and nonconsciously affected the way in which they behaved toward White and Black applicants. Although the interviewers did not consciously realize it, they behaved in subtly more negative ways toward the Black applicants than they did toward the White applicants. They spoke more quickly to African Americans, avoided eye contact, and sat at greater distances. In the end, it was clear that candidates who were treated in these ways performed objectively worse in their job interviews (Word et al., 1974).

These studies, as well as many others, should serve as a caution to parents and teachers. Believing that girls dislike computers or are not competent with them can directly lead to girls being treated differently when interacting with computers. Adult educators and parents who introduce the computer tasks to children may have negatively stereotyped beliefs (like the interviewers in the Word et al. study) and communicate these beliefs in subtle ways to their children. Self-fulfilling prophecies have the added danger of promoting resiliency in the stereotype because those parents and teachers who hold stereotypical beliefs will often see them confirmed. What they do not realize is that their lessons, their examples, and their communications may have contributed to creating the very disparity they believe they are observing (Schofield, 1995).

The self-fulfilling prophecy literature also provides examples of how expectations about *oneself* can affect one's future per-

formance. Zanna, Sheras, Cooper, and Shaw (1975) modified Rosenthal and Jacobson's procedure by telling not only the teachers but also the students of the results of a bogus ability test. As was the case when the teachers believed certain students would excel, students who had been told that they should improve dramatically during the year drastically outperformed their peers in reading and math at year's end. In the context societal messages, a girl who comes to believe the stereotype about her gender can be expected to give up more easily and not become as competent as her peers. This link between computer attitudes and ability makes computer anxiety part of a vicious and mutually reinforcing cycle.

Boys, in believing that technology is in their domain, are likely to benefit from the self-fulfilling prophecy. Their parents, teachers, and other socializing agents act in ways that produce positive feelings about computers, and the boys may respond positively in the way they approach, think about, and perform with computers. This is also a reinforcing cycle. When trying to dismantle the negative cycle that holds back girls, one cannot neglect the positive cycle that enables boys. The message should be that technology is everyone's domain.

Mere Awareness: Stereotype Threat

People can also believe that the stereotype exists but disbelieve in the truth of the stereotype. For example, Devine's (1990) work on racial stereotypes shows that, regardless of whether White participants were prejudiced or not, they could quickly and automatically produce the list of traits that form Whites' stereotypes of Blacks. Even the most self-confident female computer scientist cannot help but be aware that most of the population considers her an anomaly. Unfortunately, even the mere knowledge of a stereotype can have harmful effects. This is another path between the gender stereotype and computer anxiety. It is called "stereotype threat."

Research on stereotype threat has shown that the mere knowledge of a negative stereotype applying to one's group can cause one to perform poorly at a particular task (Spencer, Steele, & Quinn, 1999; Steele, 1997; Steele & Aronson, 1995). Therefore, how does a girl in an introductory programming course get harmed by a stereotype she doesn't believe? She feels that girls are just as good at computers as boys. Or maybe she thinks that, even if the stereotype is true about girls in general, it is not true about her. She likes to use computers and generally does very well with them. Where is the weakness?

She could have two worries that would cause her to feel anxious. First, she may worry that others will judge her based on the stereotype. Even though she is bright and accomplished, she may worry that others still view her as a stereotypical woman. She may still have her work questioned and belittled just because of her sex (Nicholson et al., 1998; Schofield, 1995). It is a worry common to groups subject to discrimination. Such worries can distract her from the task at hand. With cognitive resources devoted to worrying about whether she is being judged according to the gender stereotype, the student may perform less well at the task.

The other worry is linked to the first. She knows there is a stereotype and she wants to disprove it. She doesn't just want to be good; she wants to overcome the deficit that people might believe exists. Task importance has been shown to be a key variable in stereotype threat research. The more something matters, the more it hurts to be reminded of the stereotype.

As in much literature about stereotypes, work on stereotype threat began with studies involving African Americans. Given the stereotype that African-American students are not as academically capable as White students, Steele and Aronson (1995) predicted that making this stereotype relevant would decrease the performance of black subjects taking a standardized test. Initially, they made the test relevant by saying that the test was a very reliable measure of academic ability. And students who thought the test was relevant performed worse than those who did not. The researchers went on to show that making the stereotype more salient also worked even if they did not tell the test-takers that the exam should accurately reflects their intelligence. Think about the two worries in the context of these experiments. Steele and Aronson's (1995) subjects were so concerned about confirming the stereotype that they did precisely that.

Spencer, Steele, and Quinn (1999) showed a similar effect using gender differences in mathematics. When women were told that the results of a test showed gender differences, they performed more poorly than when gender differences were not mentioned. Interestingly, the experimenters did not actually say that men outperformed women in the "gender-differences" conditions, but all subjects came to the appropriate conclusion.

Other research has shown that stereotype threat can affect a wide array of activities. Stone and colleagues have shown that Whites are likely to experience stereotype threat on activities that are alleged to be indicants of "natural athletic ability" when compared to Blacks (Stone, Lynch, Sjomeling, & Darley, 199l; Stone, Perry, & Darley, 1991). To revisit the math finding, Aronson and his colleagues showed that stereotype threat affected White males' performance on a mathematics test when they thought that the test was diagnostic of their performance relative to that of Asian-American males (Aronson, Lustina, Good, Keough, Steele, & Brown, 1999). Stereotype threat is a pervasive phenomenon that works by increasing anxiety and cognitive load.

The stereotype that links gender to computer performance is as well known as the others. Although there are currently no published studies to link the stereotype threat experienced by women to performance on the computer, the link between the two is clear to see. Especially when a student identifies with her gender and gender is a salient aspect of the social situation in which the student finds herself, then the same threat that dealt a blow to women's performance on mathematics tests. White males' performance on an athletic task should occur to women using the computer. Simply through knowledge of the existence of the stereotype, the woman is more likely than the man to succumb to that stereotype demonstrating greater anxiety and poorer performance.

Compounding the Problem: Gender Differences in Attribution

One of the more ubiquitous assumptions in the behavioral sciences is that people strive to understand the causes of their behavior in the social world. This axiom has given rise to a field of study known as "attribution" (Heider, 1958; Jones & Davis, 1965;

Kelley 1967; Weiner, 1979). In the wake of witnessing behavior, people are motivated to make attributions for the causes of the act, whether it is their own act or the behavior of another. If an athlete catches a football on the gridiron, was it because he was a good athlete (an internal attribution) or because it was an easy pass to catch (an external attribution) or simply because he was lucky, opening his arms at the right moment with the ball floating into his hands at just the right moment? Similarly, if I answer a question correctly on a standardized test, I wonder if it was because I am smart, because the test question was easy, or because I luckily picked the correct answer from a set of equally obscure alternatives.

In achievement domains that are stereotypically male, a pattern of attributions occurs that is protective for boys but damaging for girls. Boys come to feel that any success they achieve at a stereotypically male task is a function of their ability, whereas any failure is due to lack of trying, bad luck, or an unduly difficult task. This is a protective pattern because success serves to bolster boys' opinions of how good they are. It enables them to complete even more difficult tasks with a strong belief in their own ability to succeed. If they don't succeed, they can rely on the notion that they only need to try harder, pay more attention, or be more judicious in their choice of tasks. Lack of success, in short, does not translate into a belief in lack of ability.

Girls, on the other hand, make a very different pattern of attribution. Success is attributed moreso because of external factors such as luck, effort, or an unduly easy task; whereas failure is taken personally as confirmation of a lack of ability. This potentially damaging attributional pattern causes girls to believe that the primary route to success in a stereotypically male domain like math, science, or computer technology is through luck or effort. By continuously working hard, a girl may feel that she can achieve success, but it is not because of her ability or intelligence at the task. Failure, by contrast, can be devastating because it can provide evidence for what she already believed: That she is not a capable performer in the world that has, according to stereotypes, been the province of boys (Diener & Dweck, 1978; Licht & Dweck, 1984; Nicholls, 1975).

Parsons and colleagues examined attributional differences in the field of mathematics (Parsons, Meece, Adler, & Kaczula, 1982). Children were asked to rate their own ability at mathematics, as well the difficulty level of the courses they were taking. The children's parents were also asked to rate their children's math ability and course difficulty. Parents of girls believed that their children had less mathematical ability than did parents of boys. Moreover, the girls' view of their own ability was related to their parents' view, but not to their actual performance in math classes. Objective school records showed neither overall difference in the children's performance nor any difference in the difficulty of the courses. Nonetheless, parents unintentionally socialized their children into thinking that the girls were less gifted in math than their male counterparts.

More recently, Tiedermann (2000) surveyed several hundred elementary-school students and their parents concerning the children's ability in mathematics. As in the Parsons et al. (1982) study, there were no objective differences between boys and girls as measured by their school records. Tiedermann found that both mothers and fathers thought that boys were more skilled in mathematics than girls. And the more strongly parents believed in the gender stereotype about math, the more they attributed greater mathematics ability to their sons, but not to their daughters.

The consensus of studies that have been conducted in the attribution tradition shows that boys and girls do make different attributions for success and failure in stereotypically male domains and that the impact on girls' confidence in their own ability is damaging. Nelson and Cooper (1997) adopted these insights and applied them to the field of information technology. Ten-year-old boys and girls were asked to unscramble anagrams on a computer. After a few trials, half of the boys and half of the girls began to see error messages appear on the screen. The error messages increased in frequency and severity until they finally stated that the computer was shutting down and the drive was about to be destroyed. The other half of the children received no such error messages and, in the end, reported that no computer errors had been detected.

Nelson and Cooper (1997) then asked the children to indicate what they thought their ability level was at computer tasks. Following a successful performance (i.e., without error messages), girls thought that their ability was about average compared with other 10-year old girls, In the failure message condition, however, they thought their ability was significantly inferior to the average 10-year old. Boys thought their computer ability was higher than the average 10-year old, and failure messages did not affect their confidence in their ability. In addition, when the boys and girls in the error-message (failure) condition were asked to describe the reason for their failure to complete the task, girls were three times more likely to attribute the failure to their lack of ability than were boys. In the success condition, boys were much more likely than girls to attribute their smooth and errorless performance to their own ability, while girls were more likely to evenly distribute their attributions to good luck, persistent effort, or an easy task.

Working with college students in Germany, Dickhauser, and Stiensmeyer-Pelster (2002) also asked students to make attributions for success and failure at a computer task. Like the young children in the Nelson and Cooper (1997) study, university males were much more likely to attribute failure to a defective computer; whereas females were more likely to attribute failure to their own ability, causing females to feel greater shame about their performance and lowered expectation about future interactions with the computer.

The conclusion of these studies points to the fact that children are taught by their parents about how to make attributions for success and failure. At least in the academic areas that are stereotypically seen as male domains, parents teach their children that success for boys is due to ability, but success for girls is due to more ephemeral factors that do not rely on girls' internal capabilities. In contrast, girls' failure is an indictment of their capability at the task but boys need only work harder or have better luck in order to achieve in the future. These attributions have consequences. Boys are encouraged to keep trying, because they have the basic ability to succeed. Girls, on the other hand, are not stimulated by success because success is not a reflection of their ability. Failure is. Therefore, it is not surprising that, in the last step of the Nelson and Cooper (1997) study, children were asked if they would like to try another task either on the computer or by traditional paper and pencil. Following failure at the anagrams task, girls were much less likely than boys to want to interact with a computer for the future task.

DISMANTLING THE DIVIDE

The digital divide has made it difficult for women to participate fully in the technological age. As we have seen, women now use computers frequently but continue to feel greater anxiety, more negative attitudes, and lower personal efficacy in their interactions with computer technology. The cycle is continuous. Beginning with the shared knowledge, or stereotype, that computers are the province of men and boys, women and girls either succumb to that stereotype or fight against it. Either way, the mere knowledge that the stereotype exists causes girls and women to experience anxiety. They need not believe the stereotype is true; nor do they need to believe that the stereotype applies to them. This computer anxiety often leads to negative attitudes and lowered performance, which is then interpreted via attributional processes to reflect the accuracy of the stereotype. Parents and educators inadvertently teach girls to attribute any success they may have with computer technology to luck or effort, which limits the ameliorative role that positive performance can have on a girl's estimation of her capability to succeed at computer tasks. The attributional patterns that boys develop allow them to benefit from successful performance and shield them from being dissuaded or discouraged by the occasional occurrence of errors and failures.

Software Design Can Limit the Gender Stereotype

How then can the cycle be disrupted? One place to begin is with the stereotype itself. It cannot be willed out of existence overnight. However, its pervasiveness can be disrupted in a number of ways. Educators and software designers were partially insightful when they saw the opportunity to use technology to make learning fun. With the power of digital technology to accommodate children's varied interests and fantasies, learning can be placed in a context that children find meaningful. What the designers did not see clearly was that their collective decision to model educational software on the image of the video arcade set in motion a series of psychological and sociological factors that helped to reify the image of computers as being the province of boys—i.e., the people who are fascinated by the video arcade.

The design of educational software needs to change. This has already begun to happen, with far more of a variety of educational programs available in the current decade than existed in the past two decades. Nonetheless, the pace of educational software that appeals equally to both genders needs to quicken. The software, along with the various peripheral interfaces, is a significant communicator of the computer stereotype. As we have seen, the educational and entertainment software packages available at the end of the 20th century were communication packages directed at boys. They were written as though educators were speaking directly to boys, encouraging them to learn by having fun, and simultaneously leaving girls out of the conversation.

One means to change the communication pattern of computer technology is to change the metaphor that characterizes the fantasy elements of the game. A war metaphor is a strong communication that that the learning technology is for boys; a sports metaphor is a less potent, but probably similar, communication. Learning a lesson for the purpose of hitting a home run, making a goal, or scoring a touchdown are less likely to pique the interests of most girls; nor would such metaphors serve to weaken the stereotype that educational computing has been designed for boys.

Learning what girls like, and ultimately designing information technology software that appeals to girls, is an empirical question. We need to study girls' interactions with computer technology and design software that addresses their interests and preferences. Lepper and Malone (1987) led the way by showing that girls prefer educational communication to be in the form of a learning tool. If it is to be preferred over other communications, then it much teach what needs to be learned in a direct and efficient way (Lynn et al., 2003). Moreover, the software attracts the interest of girls to the extent that it is interactive, and involves communication and sharing (Light et al., 2000; Littleton et al., 1992).

Other forms of human computer interaction (HCI) issues are also important. What kinds of peripheral devices should be used? How will the student ask for help? Will the screen be used to keep score? Will there be a score? Will lights flash and objects explode in order to increase attention, curiosity, and interest? These are not trivial considerations. When girls come to feel that the computer is not a tool intended for them, part of their belief may be through the human factors assumptions made about their interactions with the machine. Passig and Levin (1999) worked with kindergarten children to assess what they liked about computer interfaces. They found that boys, compared to the girls, preferred to use navigational buttons to discover how a game should be played. Girls preferred that writing be part of the game and preferred a way to ask for help directly rather than use computer interfaces such as buttons. Passig and Levin (1999) found that the children's satisfaction with the games, and their time-on-task for learning at the computer, were direct functions of whether the human-factor decisions were consistent with their gender preferences.

The creation of software programs that speak to both genders is a first step. A second step is to encourage parents, school boards, and educators to purchase such programs. This is a business and educational issue with obvious payoffs to society as well as the corporate bottom line. Recent, successful commercial experience with such programs as *Barbie Fashion Designer* (Subrahmanyarn & Greenfield, 1998) and *Purple Moon* attests to the financial viability of such enterprises. Whether it is more appropriate to utilize current gender stereotypes in order to be certain that there are computer programs accessible to each gender, or whether it is best to design programs that have elements that both genders enjoy is a complex issue that goes beyond the scope of the current chapter. However, the conscious and deliberate focus that ascertains that programs educators put in schools and that parents bring to the home is a necessary step to weaken the stereotype that forms the crux of the digital divide.

Decisions about software design can also be informed by examining data collected from adults. Men and women across a broad age range have logged into Massive Online Role Playing Games (MMORPGs). Men vastly outnumber women in the game (Yee, 2006a), but it is still informative to illustrate the kinds of

activities and parameters within the MMORPGs that interest women. Players can choose a character, or avatar, to represent them in the game. In one data set, 48% of the males chose to be represented by a female character; whereas only 23% of the female players chose a male avatar. This suggests the importance of having female protagonists in computer activities. Those activities, such as learning tools for school-aged children, that require an identification between player and a character on the screen, will most likely cause a female player to feel uncomfortable if she is not represented by a female protagonist. Women also are more likely then men to view their avatar as an idealized representation of themselves and to see their in-game behavior as similar to their real-life behavior (Yee, 2007).

Women are also more likely then men to use the MMORPG environment to build supportive social networks. According to (Yee, 2006b), women form stronger friendships in the game than do men and shun playing for achievement, dominance and advancement. On a percentage basis, women are more likely than men to join the game with friends and are much more likely to join with a romantic partner. They are also more likely to share contact information. Men play for power, dominance, and points. Women play to enjoy the communication, social contact, and social interaction. Games that are multifaceted such as the MMORPG environment allow for players to find elements, activities, and goals that appeal to both genders. Careful scrutiny of what appeals to women in these multifaceted games can highlight the features that would make games for young and adult females more interesting and more acceptable.

Disrupting the digital divide at the level of software stereotyping is the most important step in reducing its deleterious impact on women. The self-fulfilling nature of the stereotype and the existence of stereotype threat will cease to be issues that support the divide if the stereotypes are diminished. Software designers need to orient their software to appeal to all groups of users. This is particularly true of programs used in education because, for many children, school represents their first exposure to the computer. It is here that they learn whether computer technology is competitive, like the games in video arcades or is a medium of communication that is equally accessible to girls and boys.

Adjusting the Social Context

The social context of computing can also help to weaken the digital divide. In line with the research of Robinson-Staveley and Cooper (1990) and Cooper et al. (1990), female college students and middle-school students performed better and experienced less anxiety when computing in private than when computing in public. One suggestion from these studies is that girls should be afforded more private space when working with computer technology. Discomfort about competitive interactions, worry about ridicule (Nicholson et al., 1998), and stress about the stereotype held by others (Steele & Aronson, 1995) can exacerbate the anxiety that perpetuates the digital divide. Where possible, structuring the work and school environments to allow private computer interactions may be quite helpful.

Another strategy for diminishing the gender divide in computing is to consider an educational structure in which girls have an opportunity to engage in computer instruction with other girls rather than boys. This can be accomplished by single-sex schools or by classes within a traditional school that are for girls only. Research with young children (Light et al., 2000) showed that girls performed significantly better in same-sex dyads. Jackson and Smith (2000) applied the single-gender concept to 11–13-year-old girls in their math classes. They found evidence for increased performance and lower stress when the girls took their class in the single-gender format.

Changing the Conclusions

Parents and educators are prone to succumb to the same stereotype as the rest of society: If computer technology is an area that boys do well in, then it must be true that girls' success is a lucky break or a matter of sheer determined effort. In our roles as parents and educators, we must be vigilant about the two elements of that trap. First, it is only a stereotype that supports the first premise: Girls are not intrinsically worse at interacting with computers than boys. Second, girls and boys must be allowed to use the outcomes of their behavior to make judgments about how good they are at computer tasks. If a child does well at computer tasks, it should be taken as evidence for his or her ability. However, research has shown that girls are not given the opportunity to benefit from their success. By unwittingly communicating attributional patterns that differ by gender, parents and educators have systematically deprived girls of benefiting from their success and deprived all children of learning from their mistakes. As Fig. 38.1 depicts, the effects of successful and unsuccessful computer performance pass through an attributional filter that undermines the beneficial effects of success and exacerbates the digital divide.

References

Abdelhamid, I. S. (2002). Attitudes toward computer: A study of gender differences and other variables. *Journal of the Social Sciences, 30*, 285–316.

Alliance of Automobile Manufacturers (2006). Today's automobile: A computer on wheels. Retrieved November 1, 2006, from http://www.autoalliance.org/innovation/cpuWheels.php

American Association of University Women Educational Foundation Commission on Technology, Gender and Teacher Education (2000). *Tech-savvy: Educating girls in the new computer age.* Washington, DC: Author.

Aronson, J., Lustina, M. J., Good, C., Keough, K., Steele, C. M., & Brown J. (1999). When White men can't do math: Necessary and sufficient factors in stereotype threat. *Journal of Experimental Social Psychology, 35*, 29–46.

Atwell, P., & Battle, J. (1999). Home computers and school performance. *The information society, 15*, 1–10.

Brosnan, M. J. (1998). The impact of psychology gender, gender-related perceptions, significant others, and the introducer of technology upon computer anxiety in students. *Journal of Educational Computing Research, 18*, 63–78.

Chappel, K. K. (1996). Mathematics computer software characteristics with possible gender-specific impact: A context analysis. *Journal of Educational Computer Research, 15*, 25–35.

Colley, A., & Comber, C. (2003). Age and gender differences in computer use and attitudes among secondary school students: What has changed? *Education Research, 45,* 155–165.

Colley, A. N., Gale, M. T., & Harris, T. A. (1994). Effects of gender role identity and experience on computer attitude components. *Journal of Educational Computing Research, 10*, 129–137.

Cooper, J., & Weaver, K. D. (2003). *Gender and computers: Understanding the digital divide*. Mahwah, NJ: Erlbaum.

Cooper, J., Hall, J., & Huff, C. (1990). Situational stress as a consequence of sex-stereotyped software. *Personality and Social Psychology Bulletin, 16*, 419–429.

Dambrot, F. H., Watkins-Malek, M. A., Silling, S. M., Marshall, R. S., & Garver, J. A. (1985). Correlates of sex differences in attitudes toward and involvement with computers. *Journal of Vocational Behavior, 27*, 71–86.

Devine, P. G. (1989). Stereotypes and prejudice: Their automatic and controlled components. *Journal of Personality and Social Psychology, 56*, 5–18.

Dickhauser, O., & Stiensmeyer-Pelster, J. (2002). Learned helplessness in working with computers? Gender differences in computer-related attributions. *Psychologie in Erziehung und Unterricht, 49*, 44–55.

Diener, C. I., & Dweck, C. S. (1978). An analysis of learned helplessness: Continuous changes in performance, strategy, and achievement cognitions following failure. *Journal of Personality and Social Psychology, 36*, 451–462.

Dundell, A., & Haag, Z. (2002). Computer self-efficacy, computer anxiety, attitudes toward the Internet and reported experience with the Internet, by gender, in an East European sample. *Computers in Human Behavior, 18*, 521–535.

Farina, F., Arce, R., Sobral, J., & Carames, R. (1991). Predictors of anxiety towards computers. *Computers in Human Behavior, 7*, 263–267.

Favio, R-A., & Antonietti, A. (2002). How children and adolescents use computers to learn. *Ricerche-di-Psicologia, 25*, 11–21.

Heider, F. (1958). *The psychology of interpersonal relations*, New York: Wiley

Huff, C., & Cooper, J. (1987). Sex bias in educational software: The effect of designers' stereotypes on the software they design. *Journal of Applied Social Psychology, 17*, 519–532.

Jackson, C., & Smith, I.D. (2000). Poles apart? An exploration of single-sex and mixed-sex educational environments in Australia and England. *Educational Studies, 26*, 409–422.

Jones, E. E., & Davis, K. E. (1965). From acts to dispositions: the attribution process in social psychology, in L. Berkowitz (Ed.), *Advances in experimental social psychology* (Volume 2, pp. 219–266), New York: Academic Press.

Kelley, H. H. (1967). Attribution theory in social psychology. In D. Levine (Ed.), *Nebraska Symposium on Motivation* (Volume 15, pp. 192–238). Lincoln: University of Nebraska Press.

Lepper, M. R., & Malone, T. W. (1987). Intrinsic motivation and instructional effectiveness in computer-based education. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning and instruction: Cognitive and affective process analysis*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lynn, K. M., Raphael, C., Olefsky, K., & Bachen, C. M. (2003). Bridging the gender gap in computing: An integrative approach to content design for girls. *Journal of Educational Computing Research, 28*, 143–162.

Licht, B. G., & Dweck, C. S. (1984). Determinants of academic achievement: The interaction of children's achievement orientations with skill area. *Developmental Psychology, 20*, 628–636.

Light, P., Littleton, K., Bale, S., Joiner, R., & Messer, D. (2000). Gender and social comparison effects in computer-based problem solving. *Learning and Instruction, 10*, 483–496.

Littleton, K., Light, P., Joiner, R., Messer, D., & Barnes, P. (1992). Pairing and gender effects on children's computer-based learning. *European Journal of Psychology of Education, 7*, 311–324.

Merton, R. K. (1948). The self-fulfilling prophecy, *The Antioch Review, 8*, 193–210.

Mitra, A., Lenzmeier, S., Steffensmeier, T., Avon, R., Qu, N., & Hazen, M. (2001). Gender and computer use in an academic institution: Report from a longitudinal study. *Journal of Educational Computer Research, 23*, 67–84.

Mucherah, W.-M. (2003). Dimensions of classroom climate in social studies classrooms where technology is available (middle school students). *Dissertation Abstracts International Section A: Humanities and Social Sciences, 60*, 1451.

Nelson, L. J., & Cooper, J. (1997). Gender differences in children's reactions to success and failure with computers. *Computers in Human Behavior, 13*, 247–267.

Nicholls, J. G. (1975). Causal attributions and other achievement-related cognitions: Effects of task outcome, attainment value, and sex. *Journal of Personality and Social Psychology, 31*, 379–389.

Nicholson, J., Gelpi, A., Young, S., & Sulzby, E. (1998). Influences of gender and open-ended software on first graders' collaborative composing activities on computers. *Journal of Computing in Childhood Education, 9*, 3–42.

Okebukola, P. A., & Woda, A. B. (1993). The gender factor in computer anxiety and interest among some Australian high school students. *Educational Research, 35*, 181–189.

Parsons, J. E., Meece, J. L., Adler, T. F., & Kaczala, C. M. (1982). Sex differences in attributions and learned helplessness. *Sex Roles, 8*, 421–432.

Passig, D., & Levin, H. (1999). Gender interest differences with multimedia learning interfaces. *Computers in Human Behavior, 15*, 173–183.

Pinkard, N. (2005). How the perceived masculinity and/or femininity of software applications influences students' software preferences. *Journal of Educational Computing Research, 32*, 57–78.

Reinen, I. J., & Plomp, T. (1997). Information technology and gender equality: A contradiction in terms? *Computers and Education, 28*, 65–78.

Robinson-Staveley, K., & Cooper, J. (1990). Mere presence, gender, and reactions to computers: Studying human-computer interaction in the social context. *Journal of Experimental Social Psychology, 26*, 168–183.

Rosenthal, R., & Jacobson, L. (1968). *Pygmalion in the classroom: Teacher expectation and pupil's intellectual development*. New York: Holt, Rinehart, Winston.

Schofield, J. W. (1995). *Computers and classroom culture*. Cambridge, MA: Cambridge University Press.

Schumacher, P., & Morahan-Martin, J. (2001). Gender, internet and computer attitudes and experiences. *Computers in Human Behavior, 17*, 95–110.

Snyder, M., & Swann, W. B. (1978). Behavioral confirmation in social interaction: From social perception to social reality. *Journal of Experimental Social Psychology, 14*, 148–162.

Snyder, M., Tanke, E. D., & Berscheid, E. (1977). Social perception and interpersonal behavior: On the self-fulfilling nature of social stereotypes. *Journal of Personality and Social Psychology, 35*, 656–666.

Solvberg, A. M. (2002). Gender differences in computer-related control beliefs and home computer use. *Scandinavian Journal of Educational Research, 46*, 409–426.

Spencer, S. J., Steele, C. M., & Quinn, D. M. (1999). Stereotype threat and women's math performance. *Journal of Experimental Social Psychology, 35*, 4–28.

Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist, 52*, 613–629.

Steele, C. M., & Aronson, J. (1995). Stereotype threat and the intellectual test performance of African Americans. *Journal of Personality and Social Psychology, 69*, 797–811.

Stone, J., Lynch, C. I., Sjomeling, M., & Darley, J. (1999). Stereotype threat effects on Black and White athletic performance. *Journal of Personality and Social Psychology, 77*, 1213–1227.

Stone, J., Perry, Z. W., & Darley, J. (1997). "White men can't jump": Evidence for the perceptual confirmation of racial stereotypes following a basketball game. *Basic and Applied Social Psychology, 19*, 291–306.

Subrahmanyarn, K., & Greenfield, P. (1998). Computer games for girls: what makes them play. In J. Cassell & H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and computer games* (pp. 46–71). Cambridge, MA: MIT Press..

Temple, L., & Lips, H. M. (1989). Gender differences and similarities in attitudes toward computers. *Computers in Human Behavior, 5*, 215–226.

Tiedermann, J. (2000). Parent's gender stereotypes and teachers' beliefs as predictors of children's concept of their mathematical ability in elementary school. *Journal of Educational Psy*chology, *92*, 144–151.

Todman, J., & Dick, G. (1993). Primary children and teacher's attitudes to computers. *Computers and Education, 20*, 199–203.

United States Census Bureau. (2005). Computer and Internet Use in the United States: 2003. October 2005. Washington DC: US Census Bureau. Retrieved March 23, 2007, from http://www.census.gov/prod/2005pubs/p23–208.pdf

Weil, M. M., Rosen, L. D., & Sears, D. C. (1987). The computerphobia reduction program: Year 1. Program development and preliminary results. *Behavior Research Methods, Instruments, and Computers, 19*, 180–184.

Weiner, B. (1979). A theory of motivation for some classroom experiences. *Journal of Educational Psychology, 71*, 3–25.

Whitley, B. E., Jr. (1997). Gender differences in computer-related attitudes and behavior: A metal-analysis. *Computers in Human Behavior, 13*, 1–22.

Wilder, G., Mackie, D., & Cooper, J. (1985). Gender and computers: Two surveys of computer-related attitudes. *Sex Roles, 13*, 215–228.

Wilson, K. R., Wallin, J. S., & Reiser, C. (2003). Social stratification and the digital divide. *Social Science Computer Review, 21*, 133–143.

Word, C. O., Zanna, M. P., & Cooper, J. (1974). The nonverbal mediating of self-fulfilling prophecies in interracial interaction. *Journal of Experimental Social Psychology, 10*, 109–120.

Yee, N. (2006a). The demographics, motivations and derived experiences of users of Massively-Multiuser Online Graphical Environments. *Presence: Teleoperators and Virtual Environments, 15,* 309–329.

Yee, N. (2006b). The psychology of MMORPGs: Emotional investment, motivations, relationship formation and problematic usage. In R. Schroeder and A. Axelsson (Eds.) *Avatars at work and play: Collaboration and interaction in shared virtual environments* (pp. 187–207). London: Springer-Verlag.

Yee, N. (2007). Meta-character: Character creation, gender bending and projection. *The Daedalus Project.* http://www.nickyee.com/daedalus/archissue.php

Yelland, N., & Lloyd, M. (2001). Virtual kids of the 21st century: Understanding the children in schools today. *Information Technology in Childhood Education Annual, 13*, 175–192.

Zanna, M. P., Sheras, P. L., Cooper, J., & Shaw, C. (1975). Pygmalion and Galatea: The interactive effect of teacher and student expectancies. *Journal of Experimental Social Psychology, 11*, 279–287.

·39·

INFORMATION TECHNOLOGY AND OLDER ADULTS

Sara J. Czaja and Chin Chin Lee
University of Miami Miller School of Medicine

Introduction **778**
Use of Computer Technology by Older Adults **778**
Usage Patterns and Trends 778
Computer and Technology Usage in Everyday Domains 779
Work Environments 779
Home Environments 780
Healthcare 781
Understanding the Older Computer User **782**
Who Are Today's Elderly? 782
Aging and Abilities 784
Sensory Processes 784
Motor Skills 784
Cognitive Abilities 785
Acceptance of Information Technologies by Older Adults **785**
Older Adults and the Acquisition of Computer Skills **786**
Aging and Computer Task Performance **786**
Designing Computer Systems to Accommodate Older Adults **787**
Hardware Considerations 787
Software Considerations 788
Other Important Design Considerations 789
Conclusions **789**
References **790**

INTRODUCTION

The expanding power of computers and the recent growth of information technologies such as the Internet have made it possible for large numbers of people to have direct access to an increasingly wide array of information sources and services. Network usage is exploding and new interfaces, search engines, and features are becoming available at an unprecedented rate. In 2003, about 61.8% of households in the United States owned a personal computer (PC) and approximately 54.7% of these households had access to the Internet (U.S. Department of Commerce, 2005; Fig. 39.1). In essence, use of technology has become an integral component of work, education, communication, and entertainment. Technology is also being increasingly used within the healthcare arena for service delivery, in-home monitoring, interactive communication (e.g., between patient and physician), transfer of health information and peer support. Use of automatic teller machines, interactive telephone-based menu systems, cellular telephones, and VCRs is also quite common. Furthermore, telephones, television, home security systems, and other communication devices are becoming more integrated with computer network resources providing faster and more powerful interactive services. In essence, in order to function independently and successfully interact with the environment, people of all ages need to interact with some form of technology.

Coupled with the technology explosion is the aging of the population. In 2004, persons 65 years or older represented 12.4% of the U.S. population and it is estimated that people in this age group will represent 20.6% of the population by 2050 (Fig. 39.2). In addition, the older population itself is getting older. Currently, about 44.5 million people are over the age of 75 years, and by the year 2050, almost 50 million people will be 75 years or older (National Center for Health Statistics, 2005; Fig. 39.2).

Given that older people represent an increasing large proportion of the population and will need to be active users of technology, issues surrounding aging and information technologies are of critical importance within the domain of human-computer interaction (HCI). To ensure that older adults are able to adapt to the new information environment, we need to understand (a) the implications of age-related changes in functional abilities for the design and implementation of technology systems; (b) the needs and preferences of older people with respect to the design of technology interfaces and software applications; and (c) the problems and challenges older adults confront when adopting new technologies. Although this topic has received increased attention within the research community, there are still many unanswered questions.

The intent of this chapter is to summarize the current state of knowledge regarding information technologies and older adults. Topics that will be discussed include adoption and use of information technologies by older adults, training, hardware and software design, and other issues such as privacy and trust in technology. A detailed discussion of the aging process will not be provided. There are many excellent sources of this material (e.g., Fisk, Rogers, Charness, Czaja, & Sharit, 2004; Birren & Schaie, 2001; Fisk & Rogers, 1997). Instead, a brief review brief review of age-related changes in abilities that have relevance to the design of technology systems will be presented. Finally, suggestions for design guidelines and areas of needed research will be summarized. It is hoped that this chapter will serve to motivate researchers and system designers to consider older adults as an important component of the HCI community.

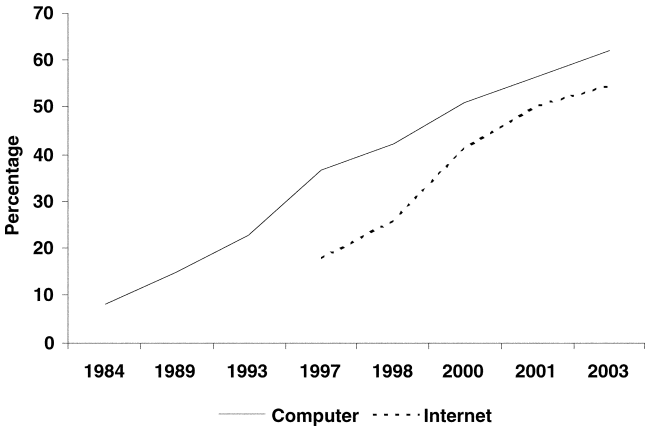


FIGURE 39.1. Trends of computer and internet access: 1984–2003. (Source: U.S. Department of Commerce, October 2005).

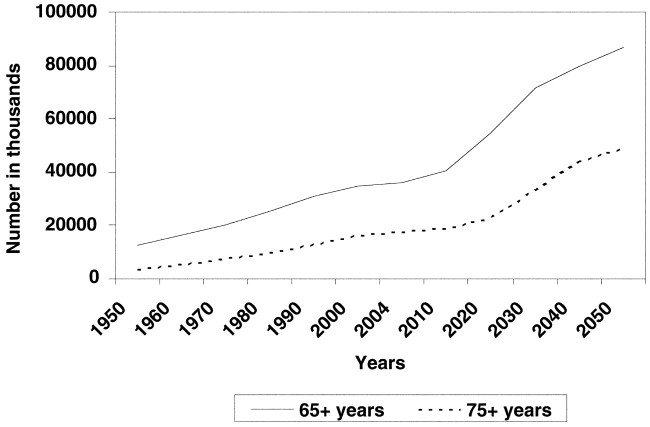


FIGURE 39.2. U.S. population. (Source: National Center for Health Statistics, 2005).

USE OF COMPUTER TECHNOLOGY BY OLDER ADULTS

Usage Patterns and Trends

As noted, there are a number of settings where older people are likely to encounter computers and other forms of communication technologies, such as the Internet, including the workplace, the home, healthcare, and service settings. However, despite a trend toward increased computer use among older people, use of computers and other forms of technology is still

lower among older people as compared to younger people. As shown in Fig. 39.3, more than 32 million of older people (55 years and older) have a computer at home as compared to 62 million of people between 35 to 54 years. Similarly although use of the Internet among older people is increasing it is still lower than that of younger age groups (Fig. 39.4). In 2005, about 26% of people age 65 and older were Internet users as compared to 67% of people age 50 to 64 and 80% of those 30 to 49 years old. Furthermore, people aged 65 and older are much less likely than younger people to have a high-speed Internet connection (Pew Internet & American Life Project, 2005). In addition, seniors who do use the Internet tend to be White, highly educated, and living in households with higher incomes (Pew Internet & American Life Project, 2004). Recent data (Czaja et al., in press) also indicated that older adults are less likely than younger adults to use other forms of technology such as automatic teller machines or VCRs and that general use of technology is a predictor of use of computers and use of the Internet.

Lack of technology use puts older adults at a disadvantage in terms of their abilities to live and function independently and to successfully negotiate the built environment. Furthermore, the full benefits of technology may not be realized by older populations. Technology holds great potential for improving the quality of life for older people. For example, computer networks can facilitate linkages between older adults and healthcare providers and communication with family members and friends, especially those who are long distant. It is quite common within the United States for family members to be dispersed among different geographic regions. In fact, nearly 7 million Americans are long-distance caregivers for older relatives (Family Caregiving Alliance, 2005). Clearly, network linkages can make it easier for family members to communicate, especially those who live in different time zones. Computers may also be used to help older people communicate with healthcare providers or other older people and may help older people become involve in continuing education. For example, telemedicine applications allow direct communication between healthcare providers and patients. There are also many opportunities to enroll in distance learning courses online. These opportunities will be more enhanced with future developments in multimedia and video conferencing technologies.

Computers and the Internet can also help older people access information about community services and resources. The Internet may also be used to facilitate the performance of routine tasks such as financial management or shopping. Access to these resources and services may be particularly beneficial for older people who have mobility restrictions or lack of transportation. Finally, many government services, such as Medicare and Social Security, have online information and are moving toward online application processes. Clearly technology is becoming an integral component of our everyday lives. The following section will present a more detailed discussion of the potential use of computer technology by older adults. This is followed by a discussion of the implications of aging for system design.

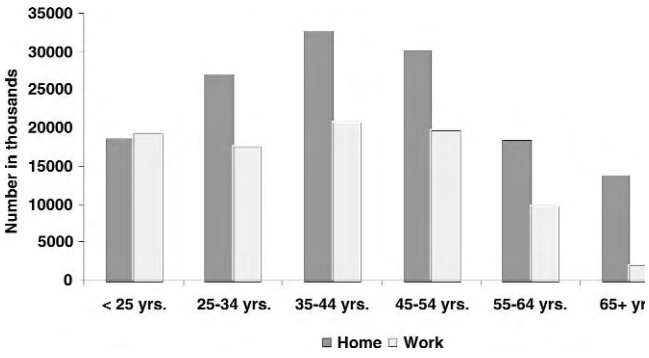


FIGURE 39.3. Percent of U.S. persons using computer. (Source: U.S. Census Bureau, Current Population Survey, October 2003).

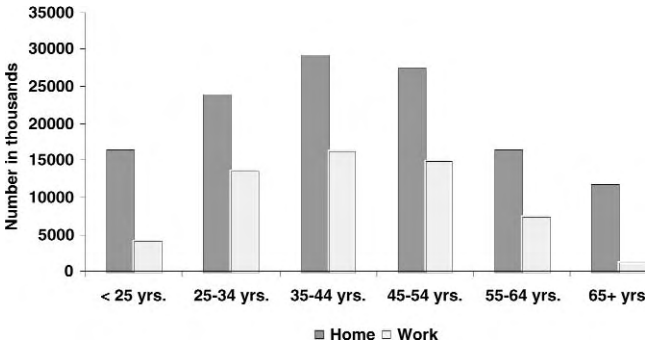


FIGURE 39.4. Percent of U.S. persons using internet. (Source: U.S. Census Bureau, Current Population Survey, October 2003).

Computer and Technology Usage in Everyday Domains

Work Environments

One setting where older people are likely to encounter computer technology is the workplace. The rapid introduction of computers and other forms of automated technology into occupational settings implies that most workers need to interact with computers simply to perform their jobs. Computer-interactive tasks are becoming prevalent within the service sector, office environments, and manufacturing industries. In 2003, 77 million people in the United States used a computer on the job these workers account for 55% of the labor force and about 2 out of every 5 employed individuals used the Internet on the job (U.S. Bureau of Labor Statistics, 2005). This is in comparison to 45% of workers in 1993 and 25% in 1984 (U.S. Census Bureau, 1999). Currently, more than 2 million adults aged 65 and older use a computer at work, and more than 1.3 million use the Internet at work (U.S. Census Bureau, 2003).

The data also suggests that technology will have a major impact on the future structure of the labor force, changing the types of jobs that are available and the way in which jobs are performed. In this regard, most of the growth will come from three occupational groups: computer and mathematical occupations, healthcare practitioners and technical occupations, and education, training, and library occupations. Other occupations that will experience growth include management and financial occupations, sales and related occupations, office and administrative

support operations, and installation, maintenance and repair occupations especially within the telecommunications industry (U.S. Bureau of Labor Statistics, 2005). Computer and communication technologies will be an integral component of most of these occupations.

The number of people who are telecommuting is also rapidly increasing. Telecommuting encompasses a number of work arrangements including home-based work, satellite office, and neighborhood telework centers. It can be done on a full- or part-time basis. In 2001, about 29 million workers in the United States engaged in some form of telecommuting, and it is estimated that there could be slightly more than 40 million telecommuters by 2010 (Potter, 2003). Telecommuting maybe particularly appropriate for older adults, as they are more likely than younger people to be mobility impaired or engaged in some form of caregiving. Telecommuting also allows for more flexible work schedules and autonomy and is more amenable to part-time work. These job characteristics are general preferred by older people. A recent study by Sharit, Czaja, Hernandez, et al. (2004) found that older people are interested in pursuing this type of work as it provides them with an opportunity to remain productive and engaged in work activities at a more flexible schedule.

However, in today's world telecommuting typically involves the use of computer and the Internet, which may be problematic for older people as they are less likely to have computer skills. The prevalence of telecommuting also raises other interesting issues such as job training. For example, can online training programs be used to train teleworkers, what is the best way to keep these workers updated on changes in job demands, and what strategies can be used to provide teleworkers with technical support? As will be discussed, issues regarding job training are particularly pertinent for older adults.

In summary, to ensure that older adults are able to adapt successfully to new workplace technologies, it is important that older adults be provided with access to retraining programs and incentives to invest in learning new skills and abilities. Greater attention also needs to be given to the design of training and instructional materials for older learners. It is also important to understand how to design technology so that it is useful and usable for older adult populations especially those with some type of impairment. Adaptive technologies may make continued work more viable for older people, especially those with some type of chronic condition or disability. For example, a number of available technologies can aid people with blindness or low vision, such as portable Braille computers, speech synthesizers, optical character recognition, screen enlargement software, and video magnifiers. Similarly, personal amplifying devices and amplified telephone receivers can aid persons with hearing loss and voice recognition software may be beneficial for persons who have limited ability to use traditional input devices, such as a mouse or keyboard because of hand or finger limitations. Clearly, a number of technologies can improve the ability of older adults to function in work environments; however, the availability of these technologies does not guarantee their success. The degree to which these technologies improve the work life of older persons depends on the usability of these technologies, the availability of these technologies within organizations, the manner in which these technologies are implemented (e.g., training), and the willingness of older people to use these devices.

Home Environments

There are a number of ways older people can use computers at home to enhance their independence and qualities of life. Home computers can provide access to information and services and can facilitate the performance of tasks such as banking and grocery shopping. Many older people have problems performing these tasks because of restricted mobility, lack of transportation, inconvenience, and fear of crime (Nair, 1989). For example, about 13% of older people report restrictions in performing activities such as shopping (Federal Interagency Forum on Aging-Related Statistics, 2004). Home computers can also be used to expand educational, recreational, and communication opportunities.

Several studies (e.g., Furlong, 1989; Eilers, 1989; Czaja, Guerrier, Nair, & Landauer, 1993) have shown that older adults are receptive to using e-mail as a form of communication and that e-mail is effective in increasing social interaction among the elderly. Increased social connectivity can so be beneficial for older people especially those who are isolated or live alone. In 2003, about 31% of all older Americans lived alone, including 41% of older women and 17% of older men (Federal Interagency Forum on Aging-Related Statistics, 2004). A recent study (Cody, Dunn, Hoppin, & Wendt, 1999) found that older adults who learned to use the Internet had more positive attitudes toward aging, higher levels of perceived social support, and higher levels of connectivity with friends and relatives.

The Internet can also be used by older people for continuing education. There are websites and software programs available on a wide variety of topics. In fact, e-learning, which refers to learning delivered or enhanced by electronic technology, is becoming one of the most popular forms of training within industry and in the education industry for the lifelong learner (Willis, 2004). There are also formal online degree programs and opportunities to be linked via videoconferencing and networking facilities to actual classrooms. The American Association for Retired Persons (AARP) has begun to offer several online courses. These learning opportunities can enable older adults to remain intellectually engaged and active especially those who have difficulty accessing more traditional classroom-based adult education programs. Research (e.g., Baltes & Smith, 1999) clearly shows that cognitive engagement and stimulation is important to successful aging. In fact, lifelong learning is a growing interest among older people. Currently, in the United States, more than 33 million adults aged 45 and older are engaged in some form of continuing education (Adler, 2002). The success of SeniorNet also points to the receptivity of older people to use computers for activities such as communication and continuing education. SeniorNet is a nonprofit organization whose mission is to provide people over the age of 50 with access to computer technology. Members learn to use computers for communicating, continuing education, financial management, and other activities, such as desktop publishing. SeniorNet also has online courses on a variety of other topics such as literature and poetry. Currently the organization has over 40,000 members and over 240 Learning Centers throughout the United States (SeniorNet, 2005).

The Internet can also create online learning communities, which bring social interaction to learning and support the learning process. An online community refers to an aggregation of

people who have a shared goal, interest, need, or activity and who have repeated interactions and share resources (Preece & Maloney-Krichmar, 2003). The imminent availability of the next generation Internet and interactive multimedia programming will further expand the education experiences that are available to individuals and enable information to be tailored to the specific needs and characteristics of users. This may be particularly beneficial to older adults who often learn at a slower pace than younger people learn and need more instructional support. A recent pilot study (Stoltz-Loike, Morrell, & Loike, 2005) found that e-learning can be an effective tool for teaching older adults technology and business related skills. The e-learning tool evaluated in the study was customized for older adults. However, as noted by the authors, the results were based on a small sample and the e-learning methods were not compared to other traditional training methods. They also pointed out that many of the e-learning materials available on the market assume a relatively sophisticated knowledge of technology and familiarity with e-learning environments. This may be disadvantageous to older people who have less knowledge of technology and less experience with computers.

In fact, despite the significant market of multimedia user interfaces in training, multimedia design "currently leaves a lot to be desired," and as with other products with which humans interface, it needs to "adopt a usability engineering approach" (Sutcliffe, 2002). Currently, there is little empirical knowledge to guide the development of these applications. In addition, almost no research has been done with older adults. This issue is especially compelling given that multimedia applications place demands on cognitive processes, such as visual search, working memory, and selective attention, which are known to decline with age.

Computers may also be used to augment the memory of older people by providing reminders of appointments, important dates, and medication schedules. Chute and Bliss (1994) showed that computers could be used for cognitive rehabilitation and for the training and retraining of cognitive skills such as memory and attention. Schwartz and Plude (1995) found that compact disc (CD) interactive systems are an effective medium for memory training with the elderly. Computers can also enhance the safety and security of older people living at home. As noted, a large proportion of older women live at home alone. Systems can be programmed to monitor home appliances, electrical, and ventilation systems and can be linked to emergency services.

Healthcare

Computer technology also holds the promise of improving healthcare for older people. Electronic links can be established between healthcare professionals and older clients, providing caregivers with easy access to their patients and allowing them to conduct daily status checks or to remind patients of home healthcare regimes. Computers may also be used for healthcare assessment and monitoring. Ellis, L. B. M., Joo, and Gross (1991) demonstrated that older people could successfully use a microcomputer-based health risk assessment. In the future computers may be used to monitor a patient's physical functioning such as measuring blood pressure, pulse rate, temperature, and so forth. The use of computers in healthcare management offers the potential of allowing many people who are at risk for institutionalization to remain at home.

The Internet is also shaping and having a pronounced impact on personal health behavior. Interactive health communication, or e-Health, generally refers to the interaction of an individual with an electronic device or communication technology (such as the Internet) to access or transmit health information or to receive or provide guidance and support on a health related issue (Robinson, Eng, & Gustafson, 1998). The scope of e-Health applications is broad but mostly encompasses searching for health information, participating in support groups and consulting with healthcare professionals. Currently, more than 70,000 websites provide health information, and in 2003, 77 million American adults searched the Internet for health information (Pew Internet & American Life, 2003). The majority of consumers search for information on a specific disease or medical problem, medical treatments or procedures, medications, alternative treatments or medicine, and information on providers or hospitals. Reasons for the growth of seeking health information online include easier access by a more diverse group of users to more powerful technologies, the development of participative healthcare models, the growth of health information that makes it difficult for any one physician to keep pace, cost containment efforts that reduce physicians time with patients, and raising concerns about self-care and prevention (Cline & Hayes, 2001).

However, the fact that consumers have access to e-Health applications has significant implications for both patients and providers. On the positive side, access to health information can empower patients to take a more active role in the healthcare process. Patient empowerment can result in more informed decision making, more tailored treatment decisions, stronger patient-provider relationships, increased patient compliance, and better medical outcomes. On the negative side, access to this wide array of health information can overload both patient and physicians, disrupt existing relationships, and lead to poor decision making on the part of consumers. For example, one major concern within the e-Health arena is the lack of quality control mechanisms for health information on the Internet. Currently, consumers can access information from credible scientific and institutional sources (e.g., Medline Plus) and unreviewed sources of unknown quality. Inaccurate health information could results in inappropriate treatment or cause delays in seeking healthcare. In fact, a recent study (El-Attar, Gray, Nair, Ownby, & Czaja, 2005) found that older adults trust health information on the Internet and generally find the Internet to be a valuable source of health information. However, they also indicated that they found health websites somewhat difficult to use. Other concerns related to the ability of nonspecialists to integrate and interpret the wealth of information that is available and the ability of healthcare providers to keep pace with their patients. Physicians increasingly report that patients come to office visits armed with information on their illness or conditions and treatment options (Ferguson, 1998). Results from an Internet user survey (Pew Internet & American Life Project, 2000) indicated that access to Internet health information has an influence on consumer decisions about seeking care, treatment choices, and their interactions with physicians. Finally, some consumers may find health information difficult to access because of

design features that result in usability problems, lack of training, or limited access to technology.

Computer technology may also be beneficial for family caregivers who are providing care for an older person with a chronic illness or disease such as dementia. Generally, the prevalence of chronic conditions or illnesses, such as dementia, diabetes, heart disease, or stroke, increases with age, and consequently, older adults (especially the "oldest old") are more likely to need some form of care or assistance. Approximately, 7 million people aged 65 years and older have mobility or self-care limitations, and about 4.5 million Americans suffer from Alzheimer's disease (AD). Family members are the primary and preferred source of help for elders. Currently, at least 22 million Americans are providing care for an adult who is ill or disabled (Family Caregiver Alliance, 2005).

Current information technologies offer the potential of providing support and delivering services to caregivers and other family members. Computer networks can link caregivers to each other, healthcare professionals, community services, and educational programs. Information technology can also enhance a caregiver's ability to access health-related information or information regarding community resources. Gallienne, Moore and Brennan (1993) found that access to a computer network, "ComputerLink", increased the amount of psychological support provided by nurses to a group of homebound caregivers of AD patients and enabled caregivers to access a support network that enabled them to share experiences, foster new friendships, and gather information on the symptoms of AD. Technology can also aid caregivers' abilities to manage their own healthcare needs as well as those of their patients by giving them access information about medical problems, treatments, and prevention strategies. Software is available on several health-related topics such as stress management, caregiving strategies, and nutrition. For example, the Alzheimer's Association has a website that has information for caregivers as does the Family Caregiving Alliance.

The Miami site of the REACH (Resources for Enhancing Alzheimer's Caregiver Health) program evaluated a family-therapy intervention augmented by a computer-telephone system (CTIS) for family caregivers of Alzheimer's patients. The intent of the CTIS system was to enhance the family therapy intervention by facilitating the caregivers' ability to access formal and informal support services caregiver. The system enabled the caregivers to communicate with therapists, family, and friends; to participate in online support groups; to send and receive messages; and to access information databases such as the Alzheimer's Association Resource Guide. A respite function was also provided. In addition, the CTIS system provided therapists with enhanced access to both the caregivers and their family members. The experience with the system was very positive with high acceptance of the system by caregivers. The majority of caregivers like the system and find it valuable and easy to use. The most common reason that caregivers used the system was to communicate with other family members, especially those who did not live nearby. The data also indicated that the system facilitated communication with other caregivers. Most caregivers reported that they found the participation in the online support groups to be very valuable (Czaja & Rubert, 2002). Finally, the data showed that caregivers who received the CTIS intervention reported less depressive symptoms especially those who were Cuban Americans (Eisdorfer et al., 2003).

Clearly, computer technology holds the promise of improving the quality of life for older adults and their families. However, for the full potential of technology to be realized for these populations, the needs and abilities of older adults must be considered in system design. As will be demonstrated in this chapter, older adults generally find technologies such as computers to be valuable and are receptive to using this type of technology. However, available data (e.g., Mead, Spauding, Sit, Meyer, & Walker, 1997; Czaja, Sharit, Ownby, Roth, & Nair, 2001) also indicated that, although older people are generally willing and able to use computers, they typically have more problems learning to use and operate computer technology than younger adults. They also have less knowledge about potential uses of computers and about how to access computers and other forms of technology. Morrell, Mayhorn, and Bennett (2000) found that the two primary predictors for not using the Internet among people aged 60 years and older were lack of access to a computer and lack of knowledge about the Internet. Other barriers to computer and Internet access include cost, lack of technical support, and usability problems. A recent study examining web usability among people aged 65 and older found that standard websites are twice as difficult to use for seniors as compared to younger adults (Nielsen Norman Group, 2005). People over the age of 65 also tend to report lower confidence than younger people in their abilities to learn to use computers (American Association of Retired People, 2002). Before, the full benefits of computer technology can be realized for older people it is important to maximize the usefulness and usability of these technologies for this population. The following section will review characteristics of older adults that have relevance to the design of computer-based systems.

UNDERSTANDING THE OLDER COMPUTER USER

Who Are Today's Elderly?

In general, older Americans today are healthier, more diverse, and more educated than previous generations (Bass, 1995). Between 1970 and 2000, the percentage of adults aged 65 and older who had completed high school increased by about 40%, and in 2000, at least 16% of people in this age group had at least a bachelor's degree. As noted, higher levels of education are typically associated with technology adoption, and people who are more educated are more likely to use computers and the Internet.

On some indices, today's older adults are healthier than previous generations. The number of people 65 and older reporting very good health and experiencing good physical functioning, such as ability to walk a mile or climb stairs, has increased in recent years. Disability rates among older people are also declining (Federal Interagency Forum on Aging-Related Statistics,

2004). However, the likelihood of developing a disability increases with age, and many older people have at least one chronic condition such as arthritis or hearing and vision impairments (Fig. 39.5).

Disability among older adults has important implications for system design. People with disabilities, especially disabled elders and minorities with disabilities, are less likely to use technology such as computers both at home and at work (Kaye, 2000; Fig. 39.6). In fact, recent data indicate that Americans of all ages who have a disability have among the lowest rates of Internet access (Pew Internet & American Life Report, 2004).

Consistent with demographic changes in the U.S. population as a whole, the older population is becoming more ethnically diverse. The greatest growth will be seen among Hispanic persons, followed by nonHispanic Blacks. Currently, individuals from ethnic minority groups are less likely to own or use technologies such as computers. This implies that technology access and training programs need to be targeted for older minority

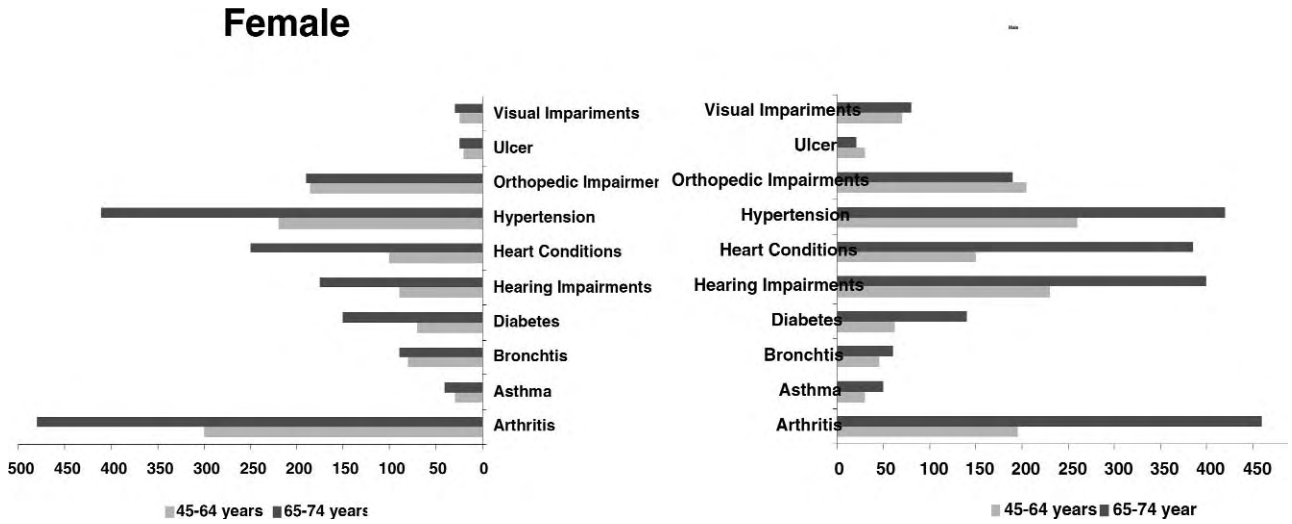


FIGURE 39.5. Chronic illness and older adults. (Source: Clinical Geriatrics, 1999).

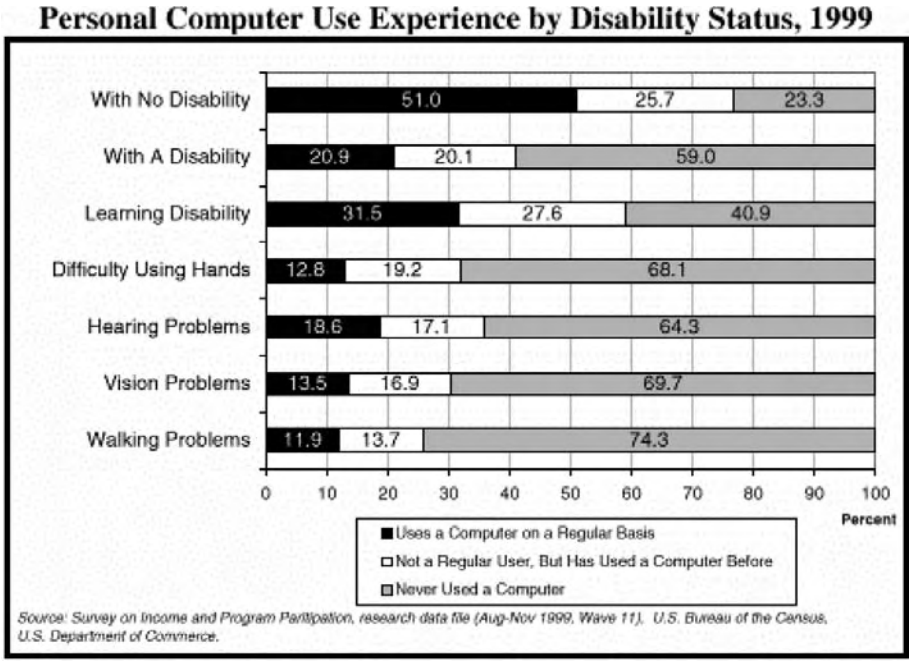


FIGURE 39.6. Computer use experience by disability status.

populations. Finally, there are more elderly women than men, and the proportion of the population that is female increases with age. In sum, health status, race, educational background, cultural traditions, and economic circumstances may all influence the adoption and use of computer-based technologies. Thus, system designers need to understand the heterogeneity of the older adult population and ensure that usability testing is done with truly representative user groups.

Aging and Abilities

Several age-related changes in functional abilities have relevance to the design of computer systems. These include changes in sensory/perceptual processes, motor abilities, response speed, and cognitive processes. In this chapter, a brief review of aging is provided to a framework for understanding the potential implications of the aging process for system design. It is important to recognize that there are substantial individual differences in rate and degree of functional change. Within any age group, young or old, there is significant variability in range of abilities, and this variability tends to increase with age.

Sensory Processes

A number of changes in visual abilities have relevance to the design of computer systems. Currently, about 14 million people in the United States suffer from some type of visual impairment, and as shown in Fig. 39.4, the incidence of visual impairment increases with age. Generally, with increased age, there is a loss of static and dynamic visual acuity. Older adults also experience a reduction in the range of accommodation, a loss a contrast sensitivity, decreases in dark adaptation, declines in color sensitivity (especially in the blue region), and heightened susceptibility to problems with glare. Visual search skills and the ability to detect targets against a background also decline with age (Kline & Schieber, 1985).

Although most older people will not experience severe visual impairments, they may experience declines in eyesight sufficient to make it more difficult to perceive and comprehend visual information. This has vast implications for the design of computer systems given that computer communications is primarily based on visually presented text. Charness, Schumann, and Boritz (1992) reported that the majority of older participants, in their studies of word processing, experienced some difficulty reading the screen and reported that these difficulties may have contributed to the lower performance of older people. In a more recent study, Charness and Holley (2001) found that older people had more difficulty than younger adults selecting small targets on a computer screen, especially when they were using a mouse as opposed to a light pen. Visual decrements may make it more difficult for older people to perceive small icons on toolbars, read e-mail, or locate information on complex screens or websites. Age-related changes in vision also have implications for the design of written instructions and computer manuals (Fisk et al., 2004)

Aging is also associated with declines in auditory acuity. Many older adults experience some decline in auditory function. Age-associated losses in hearing include a loss of sensitivity for pure tones, especially for high frequency tones; difficulty understanding speech, especially if the speech is distorted; problems localizing sounds; problems in binaural listening; and increased sensitivity to loudness (Schieber, Fozard, Gordon-Salant, & Weiffenback, 1991). These changes in audition are also relevant to design of computer systems. Older people may also find it difficult to understand synthetic speech as this type of speech is typically characterized by some degree of distortion. Multimedia systems may also be problematic for older adults. High frequency alerting sounds such as beeps or pings may also be difficult for older adults to detect.

Motor Skills

Aging is also associated with changes in motor skills including slower response times, declines in ability to maintain continuous movements, disruptions in coordination, loss of flexibility, and greater variability in movement (Rogers & Fisk, 2000). The incidence of chronic conditions such as arthritis also increases with age (Fig. 39.4). These changes in motor skills have direct relevance to the ability of older people to use current input devices such as a mouse or keyboard. For example, various aspects of mouse control such as moving, clicking, fine positioning, and dragging are likely to be difficult for older people.

A study by Smith, Sharit, and Czaja (1999) also examined age differences in the performance of basic computer-mouse control techniques (pointing, clicking, double-clicking, and dragging). The data indicated that the older participants had more difficulty performing the tasks than the younger participants especially the complex tasks such as double-clicking. Furthermore, age-related changes in psychomotor abilities such as manual dexterity were related to performance differences.

Other studies that have examined age differences in mouse performance have found similar results. For example, Riviere and Thakor (1996) found that older adults were less successful in performing tracking tasks with a mouse than were younger adults. Increased task difficulty resulted in greater age differences. Walker, Millians, and Worden (1996) compared older and younger people on a basic target acquisition task and found that older people had more difficulty especially with smaller and more distant targets. Charness, Bosman, and Elliott (1995) compared people of different ages on a target acquisition task using mice and keyboard control in one study and target acquisition and scroll-bar dragging tasks using mice and light pens in another study. Although the performance of the older people improved with practice, they performed more slowly in all cases, especially with the mouse. Findings from these studies suggest that alternative input devices might be beneficial for older people. In fact, recent data (Charness & Holley, 2001) suggested that for target selection tasks age differences in performance might be minimized with the use of alternative input devices such as a light pen. In this regard, Murata and Iwase (2005) compared target-pointing times among younger, middle-aged, and older adults using a mouse and a touch panel. They found that pointing time was longer for the older age group when using a mouse but that there were no significant age-related differences in pointing time with the touch panel. Based on these results, they recommend that for pointing tasks, a touch panel

interface should be used for middle-aged and older adults and provide design guidelines for touch panel interfaces.

Cognitive Abilities

Age-related changes in cognition also have relevance to the performance of computer-based tasks. It is well established that component behaviors comprising cognition decline with age. Processes that decline include attentional processes, working memory, discourse comprehension, problem solving and reasoning, inference formation and interpretation, and encoding and retrieval in memory (Park, 1992). Aging is also associated with declines in information processing speed. Older people tend to take longer to process incoming information and typically require a longer time to respond.

Generally computer tasks are characterized by cognitive demands. For example, component abilities such as psychomotor speed and attention are important predictors of performance of data entry tasks and menu-based tasks (Czaja & Sharit, 1998a, Sharit, Czaja, Nair, & Lee, 2003), and skills such as spatial memory and motor skills are important to word-processing performance. Similarly, searching for information on the Internet is a complex cognitive task and involves cognitive skills such as memory, reasoning, attention, learning, and problem solving. Recent data also indicated that cognitive abilities are also related to technology adoption such that people with higher fluid intelligence are more likely to adopt new technologies such as computers and the Internet (Czaja et al., in press). Given that there are age-related declines in these component abilities, use of current information technologies such as the Internet is likely to be challenging for older adults. Morrell and Echt (1996) posited that age-related declines in cognitive abilities such as working memory, processing speed, and text comprehension are influential in age-related differences in the acquisition of computer skills. They recommended that software, instructional materials, and training protocols be designed to reduce demands on these cognitive mechanisms.

Although the current technologies may present challenges for older adults, the literature generally suggests that older people are receptive to using new technologies. The following section will summarize the literature on aging and attitudes toward computers.

ACCEPTANCE OF INFORMATION TECHNOLOGIES BY OLDER ADULTS

One issue that warrants discussion when considering age and information technology is the degree to which older people are willing to interact with these types of systems and the factors that influence technology adoption. A commonly held belief is that older people are resistant to change and unwilling to interact with high-tech products. However, the available data largely disputes this stereotype. The majority of studies that have examined the attitudes of older people toward computer technology indicate that older people are receptive to using computers. However, older people do report more computer anxiety, less computer self-efficacy, and less comfort using computers than younger adults (Nair, Lee, & Czaja, 2005; Czaja et al., in press). Furthermore, computer anxiety and computer self-efficacy are important predictors of technology adoption (Ellis, E. R., & Allaire, 1999; Czaja et al., in press). However, the data also indicated that attitudes toward technology and comfort using technology are influenced by experience and the nature of interactions with computer systems as well as system design.

Several studies (e.g., Jay & Willis, 1992; Charness, Schumann & Boritz, 1992; Dyck & Smither, 1994) showed that people who had experience with computers had more positive attitudes and greater computer confidence. Other studies showed that the nature of the experience with technology influences attitudes toward that technology. For example, in a study examining age differences in acquisition of word-processing skills (Czaja, Hammond, Blascovich, & Swede, 1986), the results showed that post training attitudes were related to the training experience such that people who rated the training experience and their own performances positively also had more positive posttraining attitudes toward computers. Furthermore, there were no age effects for either pretraining or posttraining computer attitudes. Danowski and Saks (1980) also found that a positive experience with computer-mediated communication resulted in positive attitudes toward computers among a sample of older people. A more recent study (Czaja & Sharit, 1998b) examined age differences in computers as a function of computer experience among a community sample of 384 adults ranging in age from 20 to 75 years. The results indicated that, in general, older people perceived less comfort and efficacy using computers than younger people. However, experience with computers resulted in more positive attitudes for all participants irrespective of age. Kalasky, Czaja, Sharit, & Nair (1999) also found that experience with computer systems increased the comfort of older people toward using computers.

Generally, user satisfaction with a system, which is determined by how pleasant the system is to use, is considered an important aspect of usability. User satisfaction is an especially important usability aspect for systems that are used on a discretionary basis such as home computers (Nielsen, 1993). Kelley, Morrell, Park, and Mayhorn (1999) found that the most important predictor of continued use of a bulletin board system, ELDERCOMM, was success at initial training. They also found that the most positive attitudes toward computers were found among persons who used the system most often. Cody et al. (1999), in their study of Internet use among older adults, also found that people who spent the most time online had the least computer anxiety and high computer efficacy. Finally, in our study of e-mail (Czaja, Guerrier, et al., 1993), all participants found it valuable to have a computer in their home. However, the perceived usefulness of the technology was an important factor with respect to use of the technology. When the participants were asked what type of computer applications they would like available, the most common requests included emergency response, continuing education, health information, and banking/shopping.

The available data suggests that older people are not technophobic and are willing to use computers. However, the nature of their experience with technology and available technology applications are important determinants of attitudes, confidence,

and comfort using technology. Ultimately confidence and comfort influence technology adoption (Czaja et al., in press).

OLDER ADULTS AND THE ACQUISITION OF COMPUTER SKILLS

A number of studies examined the ability of older adults to learn to use technology such as computers (e.g., Elias, P. K., Elias, M. F., Robbins, & Gage, 1987; Gist, Rosen, & Schwoerer, 1988: Zandri & Charness, 1989; Czaja, Hammond, Blascovich, et al., 1986; Czaja, Hammond, & Joyce, 1989; Charness et al., 1992; Morrell, Park, Mayhorn, & Echt, 1995; Mead et al., 1997). These studies encompass a variety of computer applications and vary with respect to training strategies such as conceptual versus procedural training (Morrell, Park, et al., 1995) or active versus passive learning approaches (Czaja, Hammond, & Joyce, 1989). The influence of other variables, such as attitude toward computers and computer anxiety, on learning has also been examined.

Overall, the results of these studies indicate that older adults are, in fact, able to use technology such as computers for a variety of tasks. However, they are typically slower to acquire new skills than younger adults and generally require more help and hands-on practice. In addition, when compared to younger adults on performance measures, older adults often achieve lower levels of performance. However, the literature also indicates that training interventions can be successful in terms of improving performance, and it points to the importance of matching training strategies with the characteristics of the learner. For example, Czaja, Hammond, and Joyce (1989) found that older adults benefit from using analogies to familiar concepts and from a more active, hands-on training approach. Similarly, Mead et al. (1997) examined the effects of type of training on efficiency in a World Wide Web search activity. The participants were trained with a hands-on web navigation tutorial or a verbal description of available navigation tools. The hands-on training was found to be superior, especially, for older adults. Older adults who received hands-on training increased the use of efficient navigation tools. Mead and Fisk (1998) examined the impact of the type of information presented during training on the initial and retention performance of younger and older adults learning to use ATM technology. Specifically, they compared two types of training: concept and action. The concept training presented factual information whereas the action training was procedural in nature. The action training was found to be superior for older adults. Generally, the literature suggests that these types of strategies are also beneficial for younger people. The literature (e.g., Czaja, Sharit, Ownby, et al., 2001; Charness, Kelley, Bosman, & Mottram, 2001) also suggests that prior experience with technology is an important predictor of ability to learn to use new technology. Finally, it is important to provide older people with training on the potential uses of the technical system (e.g., what the Internet can be used for), as well as training on basic procedural operations (e.g., use of the mouse). As one would expect, the usability of the system from both a hardware and a software perspective is also important (Fisk et al., 2004).

Mayhorn, Stronge, McLaughlin, and Rogers (2004) provided suggestions for development of effective computer training for older adults. They stressed the importance of applying a systems approach to the design of training programs where the goals, abilities, and experience levels of older adults are considered in the design and evaluation of instructional programs and materials. Also, given the important role of anxiety and self-efficacy in technology adoption (Czaja et al., in press), it is important that training environments are relaxed and strategies that reduce anxiety and increase self-efficacy are incorporated into training programs.

AGING AND COMPUTER TASK PERFORMANCE

In terms of actual performance, only a handful of studies have examined the ability of older people to perform computer-based tasks that are common in work settings. For example, Czaja and Sharit (1993, 1998a, 1998b), Sharit & Czaja (1999); Czaja et al. (1998) conducted a series of studies examining age performance differences on a variety of simulated computer-based tasks (e.g., data entry, inventory management, customer service). Overall, the results of these studies indicate that older adults are willing and able to perform these types of tasks. However, the younger adults generally performed at higher levels than the older people did. Importantly, the data also indicated that there was considerable variability in performance among the older people and that with task experience those in their middle years (40 to 59 years old) performed at roughly the same levels as the young adults. In fact, task experience resulted in performance improvements for people of all ages. The results also indicated that interventions such as redesigning the screen, providing on-screen aids and reconfiguring the timing of the computer mouse improved the performance of all participants.

Other investigators examined age as a potential factor impacting on ability to use the computer for information search and retrieval. This is an important area of investigation, given that this is one of the most common reasons people use technology such as computers. For example, this type of activity is also central to use of the Internet. Also, in many work settings such as department stores, airlines, hotels, utility and health insurance companies, and educational institutions, workers are required to search through computer databases and access information to respond to customer requests. Generally, the findings from these studies indicated that, while older adults are capable of performing these types of tasks, there are age-related differences in performance. Furthermore, these differences appear to relate to age differences in cognitive abilities.

For example, Westerman, Davies, Glendon, Stammers, and Matthews (1995) examined the relationship between spatial ability, spatial memory, vocabulary skills, and age, and the ability to retrieve information from a computer database which varied according to how the database was structured (e.g., hierarchical vs. linear). In general, they found that the older subjects were slower in retrieving the information than the younger adults; however, there were no age related differences in accuracy. The learning rates also differed for the two groups such that the older people were slower than the younger peo-

ple were. They found that the slower response on the part of the older adults was dependent on general processing speed.

Freudenthal (1997) examined the degree to which latencies on an information retrieval task were predicted by movement speed and other cognitive variables in a group of younger and of older adults. The participants were required to search for answers to questions in a hierarchical menu structure. Results indicated that the older subjects were slower than the younger subjects were on overall latencies for information retrieval and that this slowing increased with each consecutive step in the menu. Similar to Westerman et al. (1995), movement speed was a significant predictor of overall latency. Other cognitive abilities, such as reasoning speed, spatial ability, and memory, were also predictive of response latencies. However, memory and spatial abilities predicted only latency on steps further into the menu structure. Freudenthal (1997) suggested that deep menu structures may not be appropriate for older adults as navigation through these types of structures is dependent on spatial skills that tend to decline with age. Vicente, Hayes, and Williges (1987) also found that age, spatial ability, and vocabulary were highly predictive of variance in search latency for a computer-based information retrieval task. They postulated that people with low spatial ability tend to "get lost" in the database. Mead, Sit, Jamieson, Rousseau, and Rogers (1996) examined the ability of younger and older adults to use an online library database. Overall, the younger adults achieved more success than did the older adults in performing the searches. They also used more efficient search strategies. The older adults also made more errors when formulating search queries and had more difficulty recovering from these errors. Czaja, Sharit, Ownby, et al. (2001), in their investigation of this type of task, which simulated a customer service representative for a health insurance company, also found that spatial skills were important predictors of performance as were response speed and prior computer experience. Kubeck, Miller-Albrecht, and Murphy (1999) investigated age differences in finding information using the Internet in a naturalistic setting. They found that older people tended to use less efficient search strategies than younger people and were less likely to find correct answers. However, they also found that, with training, older adults were successful in their searches and had very positive reactions to their Internet experiences.

As discussed, information seeking is a complex process and places demands on cognitive abilities such as working memory, spatial memory, reasoning, and problem solving. Information seeking within electronic environment also requires special skills such as knowledge related to the search system. Given that older adults typically experience declines in cognitive abilities, such as working memory, and that they are less likely than younger people to have knowledge of the structure and organization of search systems, a relevant question is the degree to which they will experience difficulty searching for information in electronic environments. Generally, the available literature suggests that older adults are able to search and retrieve information within electronic environments. However, they appear to have more difficulty than do younger adults and tend to use less efficient navigation strategies. The also appear to have problems remembering where and what they searched. In order to maximize the ability of older people to successful interact with electronic information systems, such as the Internet, and to have access to the information highway, we need to have an understanding of the source of age-related difficulties. This type of information will allow us to develop interface design and training strategies to accommodate individual differences in performance. Currently, there is very little information on problems experienced by older people when attempting to learn and navigate the Internet, especially in real-world contexts.

DESIGNING COMPUTER SYSTEMS TO ACCOMMODATE OLDER ADULTS

Hardware Considerations

As discussed, age-related changes in functioning have implications for the design of technology systems. For example, careful attention needs to be paid to the design of display screens, placement, size, shape and labeling of controls, and design and layout of instructional materials and manuals. Design features such as character size and contrast are especially important for older computer users. Larger characters and high contrast displays are generally beneficial for older people. This may not be a major problem with most computers used in the home or the workplace, as it is relatively easy to enlarge screen characters. However, it may be an issue for computers in public places such as information kiosks and ATM machines. In addition to character size and contrast, it is also important to minimize the presence of screen glare.

The organization and amount of information on a screen is important, as there are declines in visual search skills and selective attention with age. Only necessary information should be presented on a screen and important information should be highlighted. Further principles of perceptual organization, such as grouping, should be applied. Caution must also be exercised with respect to the use of color coding, as there are declines in color discrimination abilities with age. Finally, as far as possible, information should be in a consistent location and important information should stand out and be in central locations. A flaw with many existing web pages is that advertising information has more prominence than the site information (e.g., pharmaceutical advertisements on health websites).

Design and labeling of controls and input devices need special consideration. In our study of e-mail (Czaja, Guerrier, et al., 1993) people commonly confused the send and the cancel keys. Even though these keys were labeled, they were identical in size, identical in shape, and close in proximity. These findings underscore the importance of clearly differentiating controls so that labeling is easy to read and functions are easily identified.

Although there is a growing body of research examining the relative merits and disadvantages of various input devices, only a few studies have examined age effects. These studies generally suggested that commonly used input devices, such as keyboards and the mouse, might be problematic for older people. More research needs to be directed toward identifying the efficacy of alternative input devices for older people.

For example, Charness and Holley (2001) suggested that older adults may find a light pen may be easy to use because it is a direct addressing device, and it eliminates the need to translate

the target selection device onto the display. However, light pens may be difficult to use for people with arthritis or tremors as it may be hard for them to grasp and point the pen. Ellis, L. B. M., et al. (1991), in their study of a computerized healthcare system, concluded that a touchscreen version of the appraisal system may have eliminated the interface problems experienced by older users. Casali (1992) evaluated the ability of persons with impaired hand and arm functions (age unknown) and nondisabled persons to perform a target acquisition task with five cursor control devices: a mouse, trackball, cursor keys, joystick, and tablet. She found that even persons with profound disabilities were able to operate each device by using minor modifications and unique operating strategies. The mouse, trackball, and tablet resulted in better performance than the joystick and cursor keys for all participants. Consistent with Charness and Holley (2001, June), Casali found that small targets were problematic for the physically impaired users as was the task of dragging. In a follow-up study (Casali & Chase, 1993) with a sample of persons with arm and hand disabilities, the data indicated that while the mouse, trackball, and tablet tended to result in quicker performance, these devices were more error prone than the keyboard and joystick. In addition, performance improved with practice.

Clearly, much work needs to be done to evaluate which types of input devices are optimal for older people, especially for those who have restrictions in hand function. It appears that input devices, such as mice and trackballs, might not be appropriate for older adults, or they will at least require more practice to effectively use these types of devices. It may be that speech recognition devices will eliminate many of the problems (visual and movement) associated with manual input devices. In this regard, Ogozalek and Praag (1986) found that, although using a simulated listening typewriter as compared to a keyboard editor made no difference in performance of a composition task for both younger and older people, the voice input was strongly preferred by all participants. More recently, Kalasky et al. (1999) found that a commercially available speech-recognition program was robust in accepting the speech input of both younger and older adults. The technology was also acceptable to both user groups, and the data suggested that the older participants found the system more useful than their younger counterparts. These results are encouraging and suggest that speech interfaces may be especially beneficial for older people.

Software Considerations

There has been very little research examining the impact of the design of the software interface on the performance of older computer users. Given that there are age-related changes in cognitive processes, such as working memory and selective attention, it is highly likely that interface style (e.g., function keys vs. menus) will have a significant influence on the performance of older adults. The limited data that are available support this conclusion.

Joyce (1990) evaluated the ability of older people to learn a word-processing program as a function of interface style. The participants interacted with the program using one of three interface styles: on-screen menu, functions keys, and pull-down menus. She found that people using the pull-down menus performed better on the word-processing tasks. Specifically, they performed the tasks more quickly and made fewer errors. They also executed a greater number of successful editorial changes. Joyce hypothesized that the pull down-menu reduced the memory demands of the task, as the users did not have to remember editing procedures. Although the on-screen menu provided memory cues, the names of the menu items were not reflective of menu contents, thus requiring the user to remember which items were contained in which menu. The names of the pull-down menus were indicative of menu contents.

Egan and Gomez (1985) found that a display editor was less difficult for older adults that a line editor. The line editor was command based and required the user to remember command language and produce complicated command syntax. Using the display editor, changes were made by positioning a cursor at the location of change and using labeled function keys rather than a command language. Thus, fewer memory demands were associated with the display editor. Further, the display editor was less complex than the line editor, and in accordance with age-complexity hypothesis, we would anticipate smaller age effects (Cerella, Poon, & Williams, 1980).

In our study of e-mail (Czaja, Hammond, & Joyce, 1989), we found that study participants sometimes had difficulty remembering the address name required to access a particular application. We also had to provide on-screen reminders of basic commands (e.g., "Press enter after entering the address") even though the system was simple and had no complex command procedures. Similarly, in our research on text editing, we found that the older participants had more difficulty remembering editing procedures than younger adults did. Similarly, in our study of interactive telephone menu systems (Sharit, Czaja, Nair, et al., 2003), we found that older adults experienced difficulties using the menu systems and had to use the repeat function more frequently than younger adults. They also reported frustration using the menus.

Charness and Holley (2001) examined the differential effect of a keystroke-based command, menu, and menu + icon interface on word-processing skill acquisition among a sample of younger and older adults. They found that the menu and the menu + icon interface yielded better performance for all participants. The menu conditions provided more environmental support and were associated with fewer memory demands.

Consistent with the cognitive aging literature, these results suggested that systems that place minimal demands on working memory would be suitable for older people. In addition, minimal demands should be placed on spatial abilities. In this regard, interface styles such as windows should prove beneficial. However, windows may be problematic for older adults as windows are spatially organized, and it is sometimes easy to get lost within a windows environment. Also, windows may initially increase cognitive demand as users are required to learn window operations. On-screen aids such as maps and history markers may also prove to be beneficial for older people. We found, for example, that a simple graphical aid that depicted the structure of the menu system helped older adults use interactive telephone menu systems (Sharit, Czaja, Nair, et al., 2003).

Clearly, a number of interface issues need to be investigated. At the present time, we can only offer general guide-

lines with respect to designing interfaces to accommodate the needs of older users. Table 39.1 presents a summary of these guidelines. An abundance of research needs to be carried out within this area.

Other Important Design Considerations

In addition to general guidelines for the design of training programs, hardware, and software, other issues need to be considered to ensure that the benefits of technology are maximized for older people. As noted, the older adult population is very heterogeneous in terms of culture/ethnicity, health, education, and experience with technology so it is important to take into account the wide range of abilities, needs, and desires of older users to ensure that technology can adapt to their individual differences. For example, it is important to consider different designs (e.g., languages) and approaches (e.g., advertisements) for different cultural/ethnic groups. It is also important to consider how to customize or adjust interfaces and response devices to accommodate those with varying abilities. Privacy issues are also an important concern as the use of technologies become more widespread for health and financial and service applications. The issue of trust also needs consideration. Trust and ultimately acceptance of technology may be weakened by unreliability or excessive complexity. There are also potential problems with overtrust, which can be serious if technology fails. Issues of safety and maintainability must also be addressed. Finally, the cost of systems needs to be considered not only in terms of finances but also in terms of effort required on the part of the user with respect to technology access and learning and maintenance requirements.

TABLE 39.1. Summary of Interface Design Guidelines for Older Adults

1. Maximize the contrast between characters and screen background.
2. Minimize screen glare.
3. Avoid small targets and characters that are small (fonts <12).
4. Minimize irrelevant screen information.
5. Present screen information in consistent locations (e.g. error messages).
6. Adhere to principles of perceptual organization (e.g. grouping).
7. Highlight important screen information.
8. Avoid color discriminations among colors of the same hue or in the blue-green range.
9. Clearly label keys.
10. Maximize size of icons.
11. Use icons that are easily discriminated, meaningful and label icons if possible.
12. Provide sufficient practice on the use of input devices such as a mouse or trackball.
13. Provide sufficient practice on window operations.
14. Minimize demands on spatial memory.
15. Provide information on screen location.
16. Minimize demands on working memory.
17. Avoid complex command languages.
18. Use operating procedures that are consistent within and across applications.
19. Provide easy to use online aiding and support documentation.

CONCLUSIONS

There are many areas where older people are likely to interact with computer technology including the workplace, the home, service and healthcare settings. Current data indicates that older adults are generally receptive to using this type of technology but often have more difficulty than younger people acquiring computer skills and using current computer systems. They also report more anxiety about using technologies such as computers and express less confidence in their ability to learn to interact with these systems than do younger adults. This presents a challenge for HCI community.

For many older people, especially those who are frail, isolated, or have some type of mobility restrictions, access to technologies such as computers and the Internet holds promise of enhancing independence and qualities of life for older people by providing linkages to goods and resources, facilitating communication, and enhancing opportunities for work and lifelong learning. Technology can also enhance the delivery of health services to older adults and help them access information on health related topics. However, before the potential of technology is realized for older adults the needs, preferences, skills, and abilities of older people need to be understood by system designers.

Although research in this area has grown, there are many unanswered questions. For example, there are still many issues regarding design of input devices and interface design, such as how to best design speech recognition systems and menus help systems to accommodate older people. We also know little about the efficacy of design aids and support tools for older adults. In addition, we need more information on how to best train older adults to learn to use new technologies, and there are many questions regarding the design of online training programs and multimedia formats. Issues regarding privacy and trust in technology also represent critical areas of needed research. There are also many questions related to the Internet, such as how does access to Internet information impact on healthcare behavior and how do we best train seniors to identify and integrate the enormous amount of information that is available on the Internet that remain unanswered? We also need to examine how technology in the workplace impacts on employment opportunities and the work performance of older people. The issue of telecommuting has received little attention. In addition, we need more information on factors influencing technology adoption especially for minority elderly or those of lower education or economic status. There are also many questions related to quality of life and socialization that need to be addressed.

Many needs of older people would be amenable to technological solutions. However, what is lacking is a systematic effort to understand these needs and to incorporate them into design solutions and the marketplace (National Research Council, 2004). In essence, in order to design information systems so that they are useful and usable for older people, it is important to understand (a) why technology is difficult to use, when it is; (b) how to design technology for easier and effective use; and (c) how to effectively teach people to use and take advantage of technologies that are available. Answers to these questions will not only serve to benefit older adults but all potential users of technology systems.

References

American Association of Retired Persons. (2002). *Staying Ahead of the Curve: The AARP Work and Career Study*. Washington, DC: Author.

Adler, R. (2002). *The age wave meets the technology wave: Broadband and older Americans.* Retrieved December 9, 2005, from http://www.seniornet.org/downloads/broadband.pdf

Baltes, P. B., & Smith, J. (1999). Multilevel and systemic analyses of old age: theoretical and empirical evidence for a fourth age. In V. L. Bengtson & K. W. Schaie (Eds.), *Handbook of Theories of Aging* (pp. 153–173). New York: Springer.

Bass, S. A. (1995). *Older and active: How Americans over 55 are contributing to society*. New Haven, CT: Yale University Press.

Birren, J. E., & Schaie, K. W. (2001). *Handbook of the psychology and aging*. San Diego, CA: Academic Press.

Casali, S. P. (1992). Cursor control use by persons with physical disabilities: Implications for hardware and software design. *Proceedings of the 36th Annual Meeting of the Human Factors Society*, Atlanta, GA, 311–315.

Casali, S. P., & Chase, J. (1993). The effects of physical attributes of computer interface design on novice and experienced performance of users with physical disabilities. *Proceedings of the 37th Annual Meeting of Human Factors and Ergonomics Society*, Seattle, WA, 849–853.

Cerella, J., Poon, L. W., & Williams, D. (1980). Age and the complexity hypothesis. In L. W. Poon (Ed.), *Aging in the 1980s* (pp. 332–340). Washington, DC: American Psychological Association.

Charness, N., Bosman, E. A., & Elliot, R. G. (1995). *Senior-friendly input devices: Is the pen mightier than the mouse?* Paper presented at the 103 Annual Convention of the American Psychological Association Meeting, New York, NY.

Charness, N., & Holley, P. (2001, June). *Minimizing computer performance deficits via input devices and training.* Presentation prepared for the Workshop on Aging and Disabilities in the Information Age, John Hopkins University, Baltimore, MD.

Charness, N., Kelly, C. L., Bosman, E. A., & Mottram, M. (2001). Word processing training and retraining: Effects of adult age, experience, and interface. *Psychology and Aging, 16*, 110–127.

Charness, N., Schumann, C. E., & Boritz, G. A. (1992). Training older adults in word processing: Effects of age, training technique and computer anxiety. *International Journal of Aging and Technology, 5*, 79–106.

Chute, D. L., & Bliss, M. E. (1994). ProsthesisWare: Concepts and caveats for microcomputer-based aids to everyday living. *Experimental Aging Research, 20*, 229–238.

Cline, R. J. W., & Hayes, K. H. (2001). Consumer health information seeking on the Internet: The state of the art. *Health Education Research, 16*, 671–692.

Clinical Geriatrics. (1999.) Trend watch: Chronic illness and the aging U.S. population, *Clinical Geriactrics, 7*(8), 77.

Cody, M. J., Dunn, D., Hoppin, S., & Wendt, P. (1999). Silver surfers: Training and evaluating Internet use among older adult learners. *Communication Education, 48*, 269–286.

Czaja, S. J., Charness, N., Fisk, A. D., Hertzog, C., Nair, S. N., Rogers, W., et al. (in press). Factors predicting the use of technology: Findings from the Center for Research and Education on Aging and Technology Enhancement (CREATE). *Psychology and Aging*.

Czaja, S. J., & Rubert, M. (2002). Telecommunications technology as an aid to family caregivers of persons with dementia. *Psychosomatic Medicine, 64,* 469–476.

Czaja, S. J., Sharit, J., Ownby, D., Roth, D., & Nair, S. N. (2001). Examining age differences in performance of a complex information search and retrieval task. *Psychology and Aging, 16*, 564–579

Czaja, S. J., & Sharit, J. (1998a). Ability-performance relationships as a function of age and task experience for a data entry task. *Journal of Experimental Psychology: Applied, 4*, 332–351.

Czaja, S. J., & Sharit, J. (1998b). Age differences in attitudes towards computers: The influence of task characteristics. *The Journals of Gerontology: Psychological Sciences and Social Sciences, 53B*, 329–340.

Czaja, S. J., Sharit, J., Nair, S., & Rubert, M. (1998). Understanding sources of user variability in computer-based data entry performance. *Behavior and Information Technology, 19*, 282–293.

Czaja, S. J., & Sharit, J. (1993). Age differences in the performance of computer based work as a function of pacing and task complexity, *Psychology and Aging, 8*, 59–67.

Czaja, S. J., Guerrier, J. H., Nair, S. N., & Laudauer, T. K. (1993). Computer communication as an aid to independence for older adults. *Behavior and Information Technology, 12*, 197–207.

Czaja, S. J., Hammond, K., & Joyce, J. B. (1989). *Word processing training for older adults.* (Final report, Grant # 5 R4 AGO4647-03). National Institute on Aging.

Czaja, S. J., Hammond, K., Blascovich, J., & Swede, H. (1986). Age-related differences in learning to use a text-editing system. *Behavior and Information Technology, 8*, 309–319.

Danowski, J. A., & Sacks, W. (1980). Computer communication and the elderly. *Experimental Aging Research, 6*, 125–135.

Dyck, J. L., & Smither, J. A. (1994). Age differences in computer anxiety: The role of computer experience, gender, and education. *Journal of Education Computing Research, 10*, 239–248.

Egan, D. E., & Gomez, L. M. (1985). Assaying, isolating, and accommodating individual differences in learning a complex skill. *Individual Differences in Cognition, 2*, 174–217.

Eilers, M. L. (1989). Older adults and computer education: "Not to have the world a closed door." *International Journal of Technology and Aging, 2*, 56–76.

Eisdorfer, C., Czaja, S. J., Loewenstein, D. A., Rubert, M. P., Argüelles, S., Mitrani, V. B., et al. (2003). The effect of a family therapy and technology-based intervention on caregiver depression. *The Gerontologist, 43*, 521–531.

El-Attar, T., E., Gray, J., Nair, S., Ownby, R., & Czaja, S. J. (2005). Older adults and Internet health information seeking. *Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society,* Orlando, FL, 163–166.

Elias, P. K., Elias, M. F., Robbins, M. A., & Gage, P. (1987). Acquisition of word-processing skills by younger, middle-aged, and older adults. *Psychology and Aging, 2*, 340–348.

Ellis, E. R., & Allaire, A. J. (1999). Modeling computer interest in older adults: The role of age, education, computer knowledge, and computer anxiety. *Human Factors, 41*, 345–355.

Ellis, L. B. M., Joo, H., & Gross, C. R. (1991). Use of a computer-based health risk appraisal by older adults. *Journal of Family Practice, 33*, 390–394.

Family Caregiver Alliance. (2005). *Selected caregiver statistics.* Retrieved December 10, 2005, from http://www.caregiver.org/caregiver/jsp/content_node.jsp?nodeid=439

Family Caregiver Alliance. (2005). *Women and caregiving: Facts and figures*. Retrieved December 10, 2005, from http://www.caregiver.org/caregiver/jsp/content_node.jsp?nodeid=892

Federal Interagency Forum on Aging-Related Statistics (2004). *Older Americans 2004: Key indicators of well-being.* Washington, DC: U.S. Government Printing Office.

Fisk, A. D., & Rogers, W. (1997). *The handbook of human factors and the older adults.* San Diego, CA: Academic Press.

Fisk, A. D., Rogers, W., Charness, N., Czaja, S. J., & Sharit, J. (2004). *Designing for older adults: Principles and creative human factors approaches.* London: Taylor & Francis.

Ferguson, T. (1998). Digital doctoring: Opportunities and challenges in electronic-patient communication. *Journal of American Medical Association, 280*, 1261–1262.

Freudenthal, D. (1997). *Learning to use interactive devices; age differences in the reasoning process*. Unpublished master's thesis, Eindhoven University of Technology, Eindhoven, Netherlands.

Furlong, M. S. (1989). An electronic community for older adults: The SeniorNet network. *Journal of Communication, 39*, 145–153.

Gallienne, R. L., Moore, S. M., & Brennan, P. F. (1993). Alzheimer's caregivers: Psychosocial support via computer networks. *Journal of Gerontological Nursing, 12*, 1–22.

Gist, M., Rosen, B., & Schwoerer, C. (1988). The influence of training method and trainee age on the acquisition of computer skills. *Personal Psychology, 41*, 255–265.

Jay, G. M., & Willis, S. L. (1992). Influence of direct computer experience on older adults attitude towards computer. *Journal of Gerontology: Psychological Sciences, 47*, 250–257.

Joyce, B. J. (1990). *Identifying differences in learning to use a text-editor: the role of menu structure and learner characteristics*. Unpublished master's thesis, State University of New York, Buffalo, NY.

Kalasky, M. A., Czaja, S. J., Sharit, J., & Nair, S. N. (1999). Is speech technology robust for older population? *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society,* pp. 123–128.

Kaye, H. S. (2000). Computer and Internet use among people with disabilities. *Disability Statistics Reports (13)*. Washington, DC: U.S. Department of Education, National Institute on Disability and Rehabilitation Research. Available online at http://dsc.ucsf.edu/pdf/report13.pdf

Kelly, C. L., Morrell, R. W., Park, D. C., & Mayhorn, C. B. (1999). Predictors of electronic bulletin board system use in older adults. *Educational Gerontology, 25*, 19–35.

Kline, D. W., & Schieber, F. J. (1985). Vision and aging. In J. E. Birren & K. W. Schaie (Eds.), *Handbook of the Psychology and Aging* (pp. 296–331), New York: Van Nostrand Reinhold.

Kubeck, J. W., Miller-Albrecht, S. A., & Murphy, M. D. (1999). Finding information on the World Wide Web: Exploring older adults' exploration. *Educational Gerontology, 25*, 167–183.

Mayhorn, C. B., Stronge, A. J., McLaughlin, A. C., & Rogers, W. A. (2004). Older adults, computer training, and the systems approach: A formula for success. *Educational Gerontology, 30*, 185–204.

Mead, S. E. & Fisk, A. D. (1998). Measuring skill acquisition and retention with an ATM simulator: The need for age-specific training. *Human Factors, 40*, 516–523.

Mead, S. E., Sit, R. A., Jamieson, B. A., Rousseau, G. K., & Rogers, W. A., (1996). *Online library catalog: Age-related differences in performance for novice users.* Paper presented at the Annual Meeting of the American Psychological Association, Toronto, Canada.

Mead, S. E., Spaulding, V. A., Sit, R. A., Meyer, B., & Walker, N. (1997). Effects of age and training on World Wide Web navigation strategies. *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, Albuquerque, NM, 152–156.

Morrell, R. W., & Echt, C. V. (1996). Instructional design for older computer users: The influence of cognitive factors. In W. A. Rogers, A. D. Fisk, & N. Walker (Eds.), *Aging and skilled performance: Advances in theory and application* (pp. 241–265). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Morrell, R. W., Mayhorn, C. B., & Bennett, J. (2000). A survey of World Wide Web in middle-aged and older adults. *Human Factors, 42*(2), 175–185.

Morrell, R. W., Park, D. C., Mayhorn, C. B., & Echt, K. V. (1995). *Older adults and electronic communication networks: Learning to use ELDERCOMM.* Paper presented at the 103 Annual Convention of the American Psychological Association. New York, NY.

Murata, A., & Iwase, H. (2005). Usability of touch-panel interfaces for older adults. *Human Factors, 47*(4), 767–776.

Nair, S. N., Lee, C. C., & Czaja, S. J. (2005). Older adults and attitudes toward computers: Have they changed with recent advances in technology? *Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society,* Orlando, FL, 154–157.

Nair, S. N. (1989). *A capability-demand analysis of grocery shopping problems encountered by older adults.* Unpublished master's thesis, State University of New York, Buffalo, NY.

National Center for Health Statistics. (2005). *Health, United States, 2005 with chartbook on trends in the health of Americans.* Washington, DC: U.S. Government Printing Office.

National Research Council. (2004). *Technology for adaptive aging.* Steering Committee for the Workshop on Technology for Adaptive Aging. Richard W. Pew and Susan B. Van Hemel, editors. Board on Behavioral, Cognitive, and Sensory Sciences, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academics Press.

Nielsen, J. (1993). *Usability engineering*. New York: Academic Press.

Nielsen Norman Group. (2005). *Web usability for senior citizens*. Retrieved January 3, 2006, from http://www.nngroup.com/reports/seniors/

Ogozalek, V. Z., & Praag, J. V. (1986). Comparison of elderly and younger users on keyboard and voice input computer-based composition tasks. *Proceedings of CHI'86 Human Factors in Computing Systems,* New York, NY, 205–211.

Park, D. C. (1992). Applied cognitive aging research. In F. I. M. Crail & T. A. Salthouse (Eds.), *The handbook of aging and cognition* (pp. 449–494). Mahwah, NJ: Lawrence Erlbaum Associates.

Pew Internet & American Life Project. (2000). *The online healthcare evolution: How the web helps Americans take better care of themselves.* Retrieved November 25, 2005, from http://www.pewinternet.org/pdfs/PIP_Health_Report.pdf

Pew Internet & American Life Project (2003). *Internet health resources*. Retrieved November 25, 2005, from http://www.pewinternet.org/pdfs/PIP_Health_Report_July_2003.pdf

Pew Internet & American Life Project (2004). *Older Americans and the Internet.* Retrieved November 25, 2005, from http://www.pewinternet.org/pdfs/PIP_Seniors_Online_2004.pdf

Pew Internet & American Life Project (2005). *Digital divisions*. Retrieved November 25, 2005, from http://www.pewinternet.org/pdfs/PIP_Digital_Divisions_Oct_5_2005.pdf

Potter, E. E. (2003). Telecommuting: The future of work, corporate culture, and American society. *Journal of Labor Research, XXIV*, 73–84.

Preece, J., & Maloney-Krichmar, D. (2003). Online communities: Focusing on sociability and usability. In J. A. Jacko & A. Sears (Eds.), *The human computer-interaction handbook* (pp. 596–620). Mahwah, NJ: Lawrence Erlbaum Associates.

Riviere, C. N., & Thakor, N. V. (1996). Effects of age and disability on tracking tasks with a computer mouse: Accuracy and linearity. *Journal of Rehabilitation Research and Development, 33*(1), 6–15.

Robinson, T. N., Eng, P. K., & Gustafson, D. (1998). An evidence-based approach to interactive health communication: A challenge to medicine in the information age. *Journal of the American Medical Association, 280*, 1264–1269.

Rogers, W. & Fisk, A. (2000). Human factors, applied cognition, and aging. In F. I. M. Craik & T. A. Salthouse (Eds.), *The handbook of aging and cognition*. Mahawah, NJ: Lawrence Erlbaum Associates.

Schieber, F., Fozard, J. L., Gordon-Salant, S., & Weiffenbach, J. W. (1991). Optimizing sensation and perception in older adults. *International Journal of Industrial Ergonomics, 7*, 133–162.

Schwartz, L. K., & Plude, D. (1995). *Compact disk-interactive memory training with the elderly.* Paper presented at the 103 Annual Convention of the American Psychological Association, New York, NY.

SeniorNet. (2005). *Senior net Learning centers*. Retrieved December 5, 2005, from http://www.seniornet.com

Sharit, J., & Czaja, S. J. (1999). Performance of a complex computer-based troubleshooting task in bank industry. *International Journal of Cognitive Ergonomics and Human Factors, 41*, 389–397.

Sharit, J., Czaja, S. J. Hernandez, M., Yang, Perdomo, D., Lewis, et al. (2004). An evaluation of performance by older persons on a simulated telecommuting task. *The Journals of Gerontology: Psychological Sciences, 59B*(6), P305–P316.

Sharit, J., Czaja, S. J., Nair, S. N., & Lee, C. C. (2003). The effects of age and environmental support in using telephone voice menu systems. *Human Factors, 45*, 234–251.

Smith, N. W., Sharit, J., & Czaja, S. J. (1999). Aging, motor control, and performance of computer mouse tasks. *Human Factors, 41*(3), 389–396.

Stolz-Loike, M., Morrell, R.W., & Loike, J.D. (2005). Can e-learning be used as an effective training method for people over age 50? A pilot study. *Gerontechnology, 4*(2), 101–113.

Sutcliffe, A. (2002). Multimedia user interface design. In J. A. Jacko & A. Sears (Eds.), *The human computer-interaction handbook* (pp. 245–262). Mahwah, New Jersey: Lawrence Erlbaum Associates.

U.S. Bureau of Labor Statistics. (2005). *Computer and Internet use at work in 2003.* Retrieved January 3, 2006, from http://www.bls.gov/news.release/pdf/ciauw.pdf

U.S. Census Bureau. (1999). *Computer use in the United States*. Washington, DC: Author.

U.S. Census Bureau. (2003). *Current population survey*. Washington, DC: Author.

U.S. Department of Commerce (2005). *Computer and Internet use in the United States: 2003*. Retrieved January. 3, 2006, from http://www.census.gov/prod/2005pubs/p23-208.pdf

Vicente, K. J., Hayes, B. C., & Williges, R. C. (1987). Assaying and isolating individual differences in searching a hierarchical file system. *Human Factors, 29*, 349–359.

Walker, N., Millians, J., & Worden, A. (1996). Mouse accelerations and performance of older computer users. *Proceedings of Human Factors and Ergonomics Society 40th Annual Meeting,* Santa Monica, CA, 151–154.

Westerman, S. J., Davies, D. R., Glendon, A. I., Stammer, R. B., & Matthews, G. (1995). Age and cognitive ability as predictors of computerized information retrieval. *Behaviour and Information Technology, 14*, 313–326.

Willis, S. (2004). Technology and learning in current and future older cohorts. In R. W. Pew & S. B. Van Hemel (Eds.), *Technology for adaptive aging* (pp. 209–229). Washington, DC: The National Academies Press.

Zandri, E., & Charness, N. (1989). Training older and younger adults to use software. *Educational Gerontology, 15*, 615–631.

·40·

HCI FOR KIDS

Amy Bruckman, Alisa Bandlow, and Andrea Forte
Georgia Institute of Technology

Designing For and With Children **794**
How Are Children Different? **794**
Dexterity 795
Speech 796
Reading 796
Background Knowledge 796
Interaction Style 796
Children and the Design Process **797**
Use of Video With Children 797
Methods for Designing and Testing With Kids 797
Cooperative Inquiry 798
Adapting Usability Evaluation Methods 800
Traditional usability testing 800
Think/talk aloud 800
Post-task interviews 801
Codiscovery 801
Peer tutoring 801
Genres of Technology For Kids **801**
Entertainment 801
Education 802
Genres of Educational Technology 803
Computer as Tutor 803
Computer as Tool 804
Computer as Tutee 804
Computer-Supported Collaborative Learning (CSCL) 805
Technological samba schools 806
Child Safety Online 806
Conclusion **806**
References **806**

DESIGNING FOR AND WITH CHILDREN

How is designing computer software and hardware for kids different from designing for adults? Many researchers have addressed questions about the impact of technology on children; less has been said about the impact children can have on the design of technology. Methods for designing with and for children are only recently becoming widespread features of the design literature (see Jensen & Skov, 2005).

In designing for children, people tend to assume that kids are creative, intelligent, and capable of great things if they are given good tools and support. If children cannot or do not care to use technologies we have designed, it is our failure as designers. These assumptions are constructive, because users generally rise to designers' expectations. In fact, the same assumptions are useful in designing for adults. Designers of software for children start at an advantage because they tend to believe in their users. However, they may also be at a disadvantage because they no longer remember the physical and cognitive differences of being a child.

In this chapter, we will

- Describe how children's abilities change with age, as it relates to HCI
- Discuss how children differ from adults cognitively and physically, for those characteristics most relevant for HCI
- Discuss children as participants in the design process
- Review recommendations for usability testing with kids
- Review genres of computer technology for kids, and design recommendations for each genre

HOW ARE CHILDREN DIFFERENT?

As people develop from infants to adults, their physical and cognitive abilities increase over time (Kail, 1991; Miller, L. T. & Vernon, 1997; Thomas, 1980). The Swiss psychologist Jean Piaget was a leading figure in analyzing how children's cognitions evolve (Piaget, 1970). Piaget showed that children do not just lack knowledge and experience but also fundamentally experience and understand the world differently than adults. He divided children's development into a series of stages:

- Sensorimotor (birth to 2 years)
- Preoperational (ages 2 to 7)
- Concrete Operational (ages 7 to 11)
- Formal Operational (ages 11 and up; Piaget, 1970, pp. 29–33)

Contemporary research recognizes that all children develop differently, and individuals may differ substantially from this typical picture (Schneider, 1996). However, this general characterization remains useful.

In the sensorimotor stage, children's cognitions are heavily dependent on what their senses immediately perceive. Software for children this young is difficult to design. Little interaction can be expected from the child. Obviously, all instructions must be given in audio, video, or animation, since babies cannot read. Furthermore, babies generally cannot be expected to use standard input devices like a mouse effectively, even with large targets.

"Reader Rabbit Toddler" (www.thelearningcompany.com) is targeted at children ages one to three. To eliminate the need for mouse clicking, the cursor is transformed into a big yellow star with room for five small stars inside it. As the mouse is held over a target, the small stars appear one at a time. When the fifth star appears, it counts as clicking on that target. If the child does click, the process simply moves faster. The only downside is the occasional unintended click on the "Go back to the main menu" icon.

In most activities in "Reader Rabbit Toddler," nearly random mouse movement will successfully complete the activity. For example, in the "Bubble Castle" activity, the child needs to rescue animals trapped in soap bubbles that are bouncing around the screen. Random mouse movements will catch the animals relatively quickly. Yet, the parent or teacher watching a child's use of the software over time will typically begin to detect patterns in that mouse movement that become more obviously intentional—the mouse moves more directly toward the bubbles with animals in them. This is a particularly well thought out interface because it mimics how young children learn language. A baby's first attempts at sounds are greeted with great enthusiasm—the child says an unrecognizable phoneme and the parents smile and say, "You said Dada! This is Dada!" Over time, the utterance begins to really sound like the child said "Dada." An initial positive reinforcement for even the most remote attempt at the target behavior puts the child on a good learning trajectory to acquiring that behavior (Holdaway, 1979).

Many examples of software and other cultural artifacts for young children are designed in accordance with adult expectations of what a child should like. There are a few noteworthy exceptions—for example, the television show Teletubbies is out of harmony with those stereotypes. Many adults find the television show bizarre and grating, but it is wildly popular with toddlers. The designers of the original BBC television series, Davenport and Wood (1997), used detailed observations of young children's play and speech in their design. Wood commented:

> Our ideas always come from children. If you make something for children, the first question you must ask yourself is, "What does the world look like to children?" Their perception of the world is very different to that of grown-ups. We spend a lot of time watching very young children: how they play; how they react to the world around them; what they say. (Davenport & Wood, 1997)

Focus groups also played an important role (BBC, 1997). Young children are so radically different from adults that innovative design requires careful fieldwork.

While toddlers' interactions with software on a standard desktop computer afford limited possibilities, specialized hardware can expand the richness and complexity of interactions. For example, "Music Blocks" (www.neurosmith.com) is recommended for ages two and up. Five blocks fit in slots in the top of a device rather like a "boom box" portable music player. Each block represents a phrase of music. Each side of the block is a different instrumentation of that musical phrase. Rearranging

the blocks changes the music. Interaction of this complexity would be impossible for two-year-olds using a screen-based interface, but is quite easy with specialized hardware. Research on alternative computer interfaces such as tangible technologies (Ishii & Ullmer, 1997; Dourish, 2001) holds great promise for novel children's interface designs (Price, Rogers, Scaife, Stanton, & Neale, 2003; O'Malley & Stanton, 2004).

In the preoperational stage (ages 2 to 7), children's attention spans are brief. They can hold only one thing in memory at a time. They have difficulty with abstractions. They cannot understand situations from other people's points of view. While some children may begin to read at a young age, designs for this age group generally assume the children are still preliterate. It is reasonable to expect children at this age can click on specific mouse targets, but they must be relatively large. Most designers generally still avoid use of the keyboard (except "hit any key" approaches).

In the concrete operational stage (ages 7 to 11), "We see children maturing on the brink of adult cognitive abilities. Though they cannot formulate hypothesis, and though abstract concepts such as ranges of numbers are often still difficult, they are able to group like items and categorize" (Schneider, 1996, p. 69). Concrete operational children are old enough to use relatively sophisticated software, but still young enough to appreciate a playful approach. It is reasonable to expect simple keyboard use. Children's abilities to learn to type grow throughout this age group. It is reasonable to expect relatively fine control of the mouse.

Finally, by the time a child reaches the formal operational stage (ages 11 and up), designers can assume the child's thinking is generally similar to that of adults. Their interests and tastes, of course, remain different. Designing for this age group is much less challenging, because adult designers can at least partially rely on their own intuitions.

Using age as a guide can be useful; however, designers should be aware that designing too young can be just as problematic as designing too old. Children are acutely aware of their own abilities; being asked to interact with technology designed for younger children can be perceived as an affront or boring (Halgren, Fernandes, & Thomas, 1995; Gilutz & Nielsen, 2002).



FIGURE 40.1. Children playing with Music Blocks.

In addition to considering the implications of cognitive development on children's abilities to use technologies, it is important to remember that different age groups differ culturally too. Understanding what is fun or interesting for a particular age group involves understanding both children's developmental abilities and children's culturally-dependent aesthetic sensibilities. Oosterholt, Kusano, and Vries (1996) suggested that designers should avoid trying to be fashionable—what is cool changes quickly—and should target a limited age range because children's abilities and sensibilities change quickly as well.

In the next sections, we will focus on several characteristics of children that are relevant for HCI research:

- Dexterity
- Speech
- Reading
- Background knowledge
- Interaction style

Dexterity

Young children's fine motor control is not equal to that of adults (Thomas, 1980), and they are physically smaller. Devices designed for adults may be difficult for children to use. Joiner, Messer, Light, and Littleton (1998) noted that "the limited amount of research on children has mainly assessed the performance of children at different ages and with different input devices" (p. 514).

Numerous studies confirm that children's performances with mice and other input devices increase with age (Joiner et al., 1998; Hourcade, 2002). Compared to adults, children have difficulty holding down the mouse button for extended periods and have difficulty performing a dragging motion (Strommen, 1994). This means that many standard desktop interface features pose problems for young users. For example, kids have difficulty with marquee selection. Marquee selection is a technique for selecting several objects at once using a dynamic selection shape. In traditional marquee selection, the first click on the screen is the initial, static corner of the selection shape (typically a rectangle). Dragging the mouse controls the diagonally opposite corner of the shape, allowing you to change the dimensions of the selected area to encapsulate the necessary objects. Dragging the mouse away from the initial static corner increases the size of the selection rectangle, while dragging the mouse toward the initial static corner decreases the size of the selection rectangle. A badly placed initial corner can make it difficult and sometimes impossible to select/encapsulate all of the objects. Berkovitz (1994) experimented with a new encirclement technique: the initial area of selection is specified with an encircling gesture and moving the mouse outside of the area enlarges it.

Kids may have trouble double-clicking, and their small hands may have trouble using a three-button mouse (Bederson, Hollan, Druin, Stewart, Rogers, & Proft, 1996). As with adults, children can use point-and-click interfaces more easily than drag-and-drop (Inkpen, 2001; Joiner et al., 1998). Inkpen (2001) noted, "Despite this knowledge, children's software is often implemented to utilize a drag-and-drop interaction style. Bringing

solid research and strong results . . . to the forefront may help make designers of children's software think more about the implications of their design choices" (p. 30).

Strommen (1998) noted that since young children cannot reliably tell their left from their right, interfaces for kids should not rely on that distinction. In his Actimates interactive plush toy designs, the toys' left and right legs, hands, and eyes always perform identical functions. More recent interactive soft toys like "Hug & Learn Baby Tad" by Leapfrog (http://www.leapfrog.com) rely in clear visual markings on left and right paws to indicate their distinct functions.

Speech

Speech recognition has intriguing potential for a wide variety of applications for children. O'Hare and McTear (1999) studied use of a dictation program by 12-year-olds and found that they could generate text more quickly and accurately than by typing. They note that dictation automatically avoids some of the errors children would otherwise make, because the recognizer generates correct spelling and capitalization. This is desirable in applications where generating correct text is the goal. If instead, the goal is to teach children to write correctly (and, for example, to capitalize their sentences), then dictation software may be counterproductive.

While O'Hare and McTear (1999) were able to use a standard dictation program with 12-year-olds, Nix, Fairweather, and Adams (1998) noted that speech recognition developed for adults will not work with very young children. In their research on reading tutors for children 5 to 7 years old, they first tried a speech recognizer designed for adults. The recognition rate was only 75%, resulting in a frustrating experience for their subjects. Creating a new acoustic model from the speech of children in the target age range, they were able to achieve an error rate of less than 5%. Further gains were possible by explicitly accounting for common mispronunciations and children's tendency to respond to questions with multiple words where adults would typically provide a one-word answer. Even with the improved acoustic model, the recognizer still made mistakes. To avoid frustrating the children with incorrect feedback, they chose to have the system never tell the child they were wrong. When the system detects what it believes to be a wrong answer, it simply gives the child an easier problem to attempt.

Reading

The written word is a central vehicle for communicating information to humans in human-computer interfaces. Consequently, designing computer technology for children with developing reading skills presents a challenge. Words that are at an appropriate reading level for the target population must be chosen. Larger font sizes are generally preferred. Bernard, Mills, Frank, and McKnown (2001) found that kids 9 to 11 years old prefer 14-point fonts over 12-point. Surprisingly, very little empirical work has been done in this area. Most designers follow the rule of thumb that the younger the child, the larger the font should be.

Designing for preliterate children presents a special challenge. Audio, graphics, and animation must substitute for all functions that would otherwise be communicated in writing. The higher production values required can add significantly to development time and cost. Likewise, visually impaired children pose unique challenges for designers. Audio, tactile, and other sensory interfaces can provide opportunities for visually and otherwise impaired children who cannot use traditional interfaces to interact with computers autonomously (McElligott & van Leeuwen, 2004).

Background Knowledge

Many user interfaces are based on metaphors (Erickson, 1990) from the adult world. Jones (1992) noted that children are less likely to be familiar with office concepts like file folders and in-out boxes. In designing an animation system for kids, Halgren et al. (1995) found many kids to be unfamiliar with the metaphor of a frame-based filmstrip and that of a VCR. It is helpful to choose metaphors that are familiar to kids, though kids often have success in learning interfaces based on unfamiliar metaphors if they are clear and consistent (Schneider, 1996).

Interaction Style

Children's patterns of attention and interaction are quite different from those of adults. Traditional task-oriented analyses of activity may fail to capture the playful, spontaneous nature of children's interactions with technology. For example, when adults are the intended users, designers take great pains to create error messages that are informative and understandable based on the assumption that users want to avoid generating the message again. Hanna, Risden, and Alexander (1997) used a funny noise as an error message and found that the children repeatedly generated the error to hear the noise. Similarly, Halgren et al. (1995) found that children would click on any readily visible feature just to see what would happen, and they might click on it repeatedly if it generated sound or motion in feedback. This behavior was causing young users to be trapped in advanced modes they did not understand. The designers chose to hide advanced functionality in drawers—a metaphor that is familiar to children.

> By hiding the advanced tools, the novice users would not stumble onto them and get lost in their functionality. Rather, only the advanced users who might want the advanced tools would go looking for more options. This redesign allows the product to be engaging and usable by a wider range of ages and abilities. (Halgren et al., 1995)

Resnick and Silverman (2005) went even further in their design principle: "Make it as simple as possible—and maybe even simpler." They warned against "functionality creep" and suggested removing advanced functionality altogether if it is not clearly required to support children's creative efforts.

Children also bring unique interaction styles to online environments; they respond to information they encounter while browsing the web in markedly different ways than adults. In a study of 55 first through fifth graders, the Nielsen Norman

Group (Gilutz & Nielsen, 2002) found that kids were often unable to distinguish between website content and advertisements. Moreover, they rarely scrolled down to find content; instead, they chose to interact with website elements that were immediately visible. When examining a new website, children were willing to hunt for links in the content by "scrubbing the screen" with the mouse instead of relying solely on visual cues (Gilutz & Nielsen, 2002).

Children are more likely than adults to work with more than one person at a single computer. They enjoy doing so to play games (Inkpen, 1997), and they may be forced to do so because of limited resources in school (Stewart, Raybourn, Bederson, & Druin, 1998). Teachers may also create a shared-computer setup to promote collaborative learning. When multiple children work at one machine simultaneously, they need to negotiate sharing control of input devices. Giving students multiple input devices increases their productivity and their satisfaction (Inkpen, 1997; Inkpen, Gribble, Booth, & Klawe, 1995; Stewart et al., 1998). Inkpen et al. (1995) compared two different protocols for transferring control between multiple input devices: give and take. In a give protocol, the user with control clicks the right mouse button to cede it to the other user; in a take protocol, the idle user clicks to take control. In one study with 12-year-olds and another with 9- to 13-year-olds, they found that girls solve more puzzles with a give protocol, but boys are more productive with a take protocol (Inkpen, 1997; Inkpen et al., 1995).

CHILDREN AND THE DESIGN PROCESS

Users play a variety of roles in HCI design processes. Visionary designers, such as Kay (1972) and Papert (1972), began considering the abilities and sensibilities of children in the design of new technologies as early as the 1970s. Today, ethnographic and participatory (Schuler & Namioka, 1993) methods are becoming increasingly common features of the human-centered design toolkit, as HCI designers attempt to deeply understand the practices and preferences of people who will be using new technologies. When designers enter the world of children, and, conversely, when children enter the laboratory, many of the traditional rules change. As we have seen, children are not just little adults; they engage with the world in fundamentally different ways. Naturally, they bring a host of social, emotional, and cognitive elements to the design process that are unfamiliar to designers who are accustomed to working with adults. In this section, we examine new and traditional methods for working with children in a human-centered design process.

Use of Video With Children

Like adults, children may change their behaviors when a video camera is present. Druin (1999) and a design team found in their early work that children tended to "freeze" or "perform" when they saw a video camera in the room. In subsequent work, Druin's team observed that the problems associated with videotaping had more to do with power relationships than with the video cameras themselves. When the children are in control of the cameras, their discomfort decreases (Alborzi et al., 2000). In addition to considering the social impact of using a camera with children, there are also technical difficulties to deal with. Druin's research team found that, even with smaller cameras, it was difficult to capture data in small bedrooms and large public spaces. The sound and speech captured in public spaces was difficult to understand or even inaudible. Finally, it was difficult to know where to place cameras because they did not know where children would sit, stand, or move in the environment. Druin recommended using multiple data sources to capture "messy" design environments with children, including note takers and participant observers in addition to videotaping. Druin also encouraged the design team to use video cameras (along with journal writing, team discussion, and adult debriefing) as a way to record their brainstorming sessions and other design activities.

Goldman-Segall (1996) explained why video data are an important part of ethnographic interviews and observations. When using video, the researcher does not have to worry about remembering or writing down every detail: "She can concentrate fully on the person and on the subtleties of the conversation." The researcher also has access to "a plethora of visual stimuli which can never be 'translated' into words in text," such as body language, gestures, and facial expressions. It is especially important to be able to review the body language of children as they interact with software. Hanna et al. (1997) stated that children's "behavioral signs are much more reliable than children's responses to questions about whether or not they like something, particularly for younger children. Children are eager to please adults, and may tell you they like your program just to make you happy." MacFarlane, Sim, and Horton (2005) suggested that both signs (behaviors) and symptoms (children's direct responses) should be used together to understand children's enjoyments of and abilities to use new technologies. Video is extremely useful in being able to study behavioral signs as the researchers may miss some important signs and gestures during the actual observation or interview.

Instead of using video in its traditional capacity for ethnographic-style observation, some researchers have attempted to capitalize on children's playful treatment of video cameras to elicit articulation about new technologies. In studies using video probes to capture domestic communication patterns, Hutchinson et al. (2003) observed that images are particularly attractive to young people as an entertaining medium for interacting and communicating. During classroom observations of children using math-learning software, Lamberty and Kolodner (2005) encouraged children to engage in "camera talk" with stationary cameras if they wished. Many of the children regularly talked to the camera. This spontaneous behavior revealed both their preferences for using the software and their developing understanding of fractions. Likewise, Iversen (2002) suggested that, by provoking children to verbalize, video cameras provide a communication link between designers and young informants, thereby enriching both the data collected and the design experience.

Methods for Designing and Testing With Kids

In this section, we review a variety of methods for designing with and for children. These methods differ dramatically in the

amount of power they grant to children. Some methods encourage us to view children as codesigners with an equal voice in determining design direction, whereas others place children in a more reactive role as evaluators or subjects in laboratory-based usability tests. In practice, designers use methods from different points on this power spectrum depending on the maturity of the project, and often move back and forth between testing with kids and open-ended exploration (Scaife, Rogers, Aldrich, & Davies, 1997).

Druin (2002) unpacked this spectrum of control by describing four different roles that children can play in the design of new technologies: user, tester, informant, and design partner. The most reactive role she described for children in design is user. As users, children interact with existing technologies and have no direct impact on the design of the technology, except in the form of recommendations for future designs. As testers, children are asked to provide feedback about technology in development so that it can be refined before it is released; however, adult designers determine the goals of the technology much earlier. As informants, children play an earlier, more active role in determining the goals and features of new technologies. When children play the role of informant, they interact directly with designers, but ultimately, the designers decide what the children need or want based on observations, interviews, or other data collection methods. Finally, Druin explains that as design partners, children are equal stakeholders in the design process. Although they may not be able to contribute to the development of the technology in equivalent ways, their expertise is equal in importance to that of other contributors to the design process.

The notion of children as design partners will be explored more fully in the section on co-operative inquiry. Methods for including children as informants and design partners borrow from the tradition of participatory design that emerged in the Scandinavian workplace. Participatory design is an "approach towards computer systems design in which the people destined to *use* the system play a critical role in *designing* it" (Schuler & Namioka, 1993, p. xi). With children, this idea is even more important: Since they are physically and cognitively different from adults, their participations in the design process may offer significant insights. Schuler wrote:

> [Participatory Design] assumes that the workers themselves are in the best position to determine how to improve their work and their work life . . . It views the users' perceptions of technology as being at least as important to success as fact, and their feelings about technology as at least as important as what they can do with it. (Schuler & Namioka, 1993, p. xi)

Empowering children in this way and including them in the design process can be difficult due to the traditionally unequal power relationships between kids and adults.

On the other end of the spectrum, methods for including children as users and testers often borrow from the traditional practices of experimental psychology. Usability testing generally takes place in a controlled setting. Sometimes a single design is tested with the goal of improving it; at other times, different design ideas might be compared to establish which ones generate more positive feedback or enable better task completion. Data collection methods like verbal protocol analysis (Ericsson & Simon, 1993) are commonly used and will be further discussed in the section on adapting traditional usability methods for kids.

Cooperative Inquiry

Druin (1999) developed a systematic approach to developing new technologies for children with children; she created new research methods that included children in various stages of the design process. This approach, called "cooperative inquiry," is a combination of participatory design, contextual inquiry, and technology immersion. Children and adults work together on a team as research and design partners. She reiterated the idea that "each team member has experiences and skills that are unique and important, no matter what the age or discipline" (Alborzi et al., 2000, p. 97).

In this model, the research team frequently observes children interacting with software, prototypes, or other devices to gain insight into how child users will interact with and use these tools. When doing these observations, both adult and child researchers observe, take notes, and interact with the child users. During these observations, there are always at least two note-takers and one interactor, and these roles can be filled by either an adult or a child team member. The interactor is the researcher who initiates discussion with the child user and asks questions concerning the activity. If there is no interactor or if the interactor takes notes, the child being observed may feel uncomfortable, like being "on stage" (Druin, 1999). Other researchers have also found that the role of interactor can be useful for members of the design team. Scaife and Rogers (1999) successfully involved children as informants in the development of ECOi, a program that teaches children about ecology. They wanted the kids to help them codesign some animations in ECOi. Rather than just having the software designer observe the children as they played with and made comments about the ECOi prototypes, the software designer took on the role of interactor to elicit suggestions directly. Through these on the fly, high-tech prototyping sessions, they learned that "it was possible to get the software designer to work more closely with the kids and to take on board some of their more imaginative and kid-appealing ideas" (Scaife & Rogers, 1999).

When working as design partners, children are included from the beginning. The adults do not develop all the initial ideas and then later see how the children react to them. The children participate from the start in brainstorming and developing the initial ideas. The adult team members need to learn to be flexible and learn to break away from carefully following their session plans, which is too much like school. Children can perform well in this more improvisational design setting, but the extent to which the child can participate as a design partner depends on his or her age. Children younger than 7 years may have difficulty in expressing themselves verbally and in being self-reflective. These younger children also have difficulty in working with adults to develop new design ideas. Children older than 10 are typically beginning to become preoccupied with preconceived ideas of the way things are supposed to be. In general, it has been found

that children ages 7 to 10 years old are the most effective prototyping partners. They are "verbal and self-reflective enough to discuss what they are thinking" (Druin et al., 1999, p. 61), and understand the abstract idea that their low-tech prototypes and designs are going to be turned into technology in the future. They also do not get bogged down with the notion that their designs must be similar to preexisting designs and products.

Through her work with children as design partners, Druin (1999, 2002) has discovered that there are stumbling blocks on the way to integrating children into the design process and to helping adults and children work together as equals. One set of problems deals with the ability of children to express their ideas and thoughts. When the adult and children researchers are doing observations, it is best to allow each group to develop its own style of note taking. Adults tend to take detailed notes, and children tend to prefer to draw cartoons with short, explanatory notes. It is often difficult to create one style of note taking that will suit both groups. Since children may have a difficult time communicating their thoughts to adults, low-tech prototyping is an easy and concrete way for them to create and discuss their ideas. Art supplies such as paper, crayons, clay, and string allow adults and children to work on an equal footing. A problem that arises in practice is that since these tools are childlike, adults may believe that only the child needs to do such prototyping. It is important to encourage adults to participate in these low-tech prototyping sessions.

The second set of problems emerges from the traditionally unequal power relationships between adults and children. In what sense can children be treated as peers? When adults and children are discussing ideas, making decisions, or conducting research, traditional power structures may emerge. In conducting a usability study, the adult researcher might lead the child user through the experiment rather than allowing the child to explore freely on his or her own. In a team discussion, the children may act as if they are in a school setting by raising their hands to speak. Adults may even inadvertently take control of discussions. Is it sensible to set up design teams where children are given equal responsibilities to those of adult designers? Getting adults and children to work together as a team of equals is often the most difficult part of the design process. It is to be expected that it may take a while for a group to become comfortable and efficient when working together. It can take up to six months for an "intergenerational design team to truly develop the ability to build upon each other's ideas" (Druin et al., 2001). To help diffuse such traditional adult-child relationships, adults are encouraged to dress casually, and there always should be more than one adult and more than one child on a team. A single child may feel outnumbered by the adults, and a single adult might create the feeling of a school environment where the adult takes on the role of teacher. Alborzi et al. (2000) started each design session with 15 minutes of snack time, where adults and children can informally discuss anything. This helps both adults and children get to know each other better as "people with lives outside of the lab" (Alborzi et al., 2000) and to improve communication within the group.

Scaife et al. (1997) identified aspects of working with children in the role of informant that require special attention. They found that, when working in pairs, children feel less inhibited about telling strange adults what they were thinking. Other researchers have also found that pairing children, especially with friends, can help ease discomfort (Dindler, Eriksson, Iversen, Ludvigsel, & Lykke-Olesen, 2005; Als, Jensen, & Skov, 2005). Scaife et al. (1997) cautioned that adults also need to become comfortable in the role of facilitator and should take care not to intervene too quickly if children's discussions wander.

In addition to the social challenges associated with mixing adults and children as equal design partners, kids do not always know how to collaborate well with one another in the first place. Because collaborating on a design project is often a novel experience for children, organizing the activities (without imposing too rigid a structure) can help create productive sessions. For example, Guha, Druin, Chipman, Fails, Simms, and Farber (2005) described a technique to support collaboration among kids and adults during cooperative inquiry sessions called "mixing ideas." First, kids generate ideas in a one-on-one session with an adult facilitator, then they work in small groups to integrate these ideas, and finally they work in larger groups until the whole group is finally working together.

Although there have been many successes in having children participate as design and research partners in the development of software, there are still many questions to be answered about the effectiveness of this approach. Scaife and Rogers (1999) attempted to address many of the questions and problems faced when working with children in their work on informant design. The first question deals with the multitude of ideas and suggestions produced by children. Children say outrageous things. How do you decide which ideas are worthwhile? When do you stop listening? The problem of selection is difficult since, in the end, the adult decides which ideas to use and which ideas to ignore. Scaife and Rogers suggested creating a set of criteria to:

> Determine what to accept and what not to accept with respect to the goals of the system . . . You need to ask what the trade-offs will be if an idea or set of ideas are implemented in terms of critical 'kid' learning factors: that is, how do fun and motivation interact with better understanding? (Scaife & Rogers, 1999, pp. 46–47)

In addition to deciding which of the children's ideas to use, there is also the problem of understanding the meaning behind what the child is trying to say. Adults tend to assume that they can understand what kids are getting at, but kid talk is not adult talk. It is important to remember that children have "a different conceptual framework and terminology than adults" (Scaife & Rogers, 1999, p. 47).

Another problem with involving children, particularly with the design of educational software, is that "children can't discuss learning goals that they have not yet reached themselves" (Scaife & Rogers, 1999, p. 30). Can children make effective contributions about the content and the way they should be taught, something which adults have always been responsible for? Adults have assumptions about what is an effective way to teach children. Kids tend to focus on the fun aspects of the software rather than the educational agenda. A mismatch of expectations may exist if kids are using components of the software in unanticipated ways. Involving children in the design and evaluation process may help detect where these mismatches occur in the software.

Adapting Usability Evaluation Methods

HCI practices have evolved to address usefulness, enjoyability, and other measures of design success; however, usability remains a fundamental concern for HCI designers. Although efficiency and task completion are often not central to kids' goals in using technology, usability problems can create barriers to achieving other goals. For example, much research done to date has focused on designing educational software, and evaluation is primarily of learning outcomes, not usability. However, usability is a prerequisite for learning. In student projects in Georgia Tech's graduate class "Educational Technology: Design and Evaluation," many student designers are never able to show whether the educational design of their software is successful. What they find instead is that usability problems intervene, and they are unable even to begin to explore pedagogical efficacy. If children cannot use educational technology effectively, they certainly will not learn through the process of using it. MacFarlane et al. (2005) found that measurements of usability and "fun" were significantly correlated in studies of educational software for science. Usability is similarly important for entertainment, communications, and other applications. Many researchers have explored the effectiveness of traditional usability methods with children. In this section, we examine comparative assessments of usability methods and review findings and recommendations.

Traditional usability testing. Several guidelines developed for work with adults become more important when applied to children. For example, when children are asked to work as testers, it is important to emphasize that it is the software that is being tested, not the participant (Rubin, 1994). Children might become anxious at the thought of taking a test, and test taking may conjure up thoughts of school. The researcher can emphasize that even though the child is participating in a test, the child is the tester and not the one being tested (Hanna et al., 1997). Rubin recommends that you show the participant where video cameras are located; let them know what is behind the one-way mirror and whether people will be watching. With children, showing them behind the one-way mirrors and around the lab gives them "a better sense of control and trust in you" (Hanna et al., 1997, p. 12).

Markopoulos and Bekker (2002) described characteristics of kids that can impact the process and outcome of usability testing:

- Children's capacity to verbalize thoughts is still developing.
- Personality may impact both kid's willingness to speak up to adults and their motivation to please authority figures.
- The capacity to concentrate is variable among kids.
- Young children are still developing the capacity for abstract and logical thinking; they may differ in cognitive ability such as remembering several items at once.
- The ability to monitor goal-directed performance develops throughout childhood and adolescence.
- Some ages may have more pronounced gender differences than others may.
- With small children, basic motor skills ability may be a barrier to effective evaluation if kids cannot use prototypes with standard input devices.

Hanna et al. (1997) developed a set of guidelines for laboratory-based usability testing with children:

- The lab should be made a little more child friendly by adding some colorful posters but avoid going overboard as too many extra decorations may become distracting to the child.
- Try to arrange furniture so that children are not directly facing the video camera and one-way mirror, as the children may choose to interact with the camera and mirror rather than the doing the task.
- Children should be scheduled for an hour of lab time. Preschoolers will generally only be able to work for 30 minutes but will need extra time to play and explore. Older children will become tired after an hour of concentrated computer use, so if the test will last longer than 45 minutes, children should be asked if they would like to take a short break at some point during the session.
- Hanna et al. (1997) suggested that you "explain confidentiality agreements by telling children that designs are 'top-secret'" (p. 12). Parents should also sign the agreements, since they will inevitably also see and hear about the designs.
- Children up to seven or eight years old will need a tester in the room with them for reassurance and encouragement. They may become agitated from being alone or following directions from a loudspeaker. If a parent will be present in the room with the child, it is important to explain to the parent that he or she should interact with the child as little as possible during the test. Older siblings should stay in the observation area or a separate room during the test, as they may eventually be unable to contain themselves and start to shout out directions.
- Hanna et al. (1997) suggested that you should "not ask children if they want to play the game or do a task—that gives them the option to say no. Instead use phrases such as 'Now I need you to,' 'Let's do this,' or 'It's time to.'"

Think/talk aloud. An important method for collecting usability data with adults is think-aloud protocols. Think-aloud protocols in HCI research are related to verbal protocol analysis methods in psychology, in which subjects are asked to describe what they are thinking about and paying attention to while they complete some set of tasks (Ericsson & Simon, 1993). In usability tests, think-aloud methods are generally used in concert with direct observation (Nielsen, 1993). Researchers who have used this method with children have observed that children may make very few comments during testing (Donker & Reitsma, 2004). In some cases, they seem to have difficulty with concurrent verbalization—verbalizing thoughts while they complete tasks. The cognitive load associated with learning and executing task itself might interfere with kids' abilities to talk about it (Hoysniemi, Hamalainen, & Turkki, 2003).

Despite potential obstacles, it has been demonstrated that verbal comments from children can play an important role in identifying usability problems. Donker and Reitsma (2004) re-

ported that, although children produced fewer comments, those few comments provided important information about the severity of usability problems that were identified by direct observation. Likewise, other work suggested that, although using think alouds with kids may result in fewer utterances than other approaches, it can be used to generate useful usability data with both older kids aged 8 to 14 (Donker & Markopoulos, 2002; Baauw & Markopoulos, 2004) and younger children aged 6 to 7 (van Kesteren, Bekker, Vermeeren, & Lloyd, 2003).

Active intervention is closely related to think-aloud protocols but involves investigators asking planned questions to encourage testers to reflect aloud on actions at specific points while completing a task. In a small comparative study with kids ages six and seven, van Kesteren et al. (2003) found that active intervention elicited the most comments when compared to think alouds, posttask/retrospection, codiscovery, peer tutoring, and traditional usability testing.

Post-task interviews. In posttask interviews, testers are asked to describe their experiences after they have finished using a new technology to complete a set of tasks. In some cases, video data are reviewed with the participant to evoke comments. This kind of retrospective verbal protocol emerged from the same tradition as think alouds (Ericsson & Simon, 1993). Van Kesteren et al. (2003) raised the question of whether young kids' limited capacities to hold in memory several concepts at once and still-developing abilities to engage in abstract thought limit their abilities to accurately recall and recount past actions. They found that some kids age six and seven were able to recall past actions and describe the ways in which their understanding changed. They note that keeping things interesting is important; children become bored with reviewing videos unless the tasks themselves are engaging to watch. In studies with kids ages 9 to 11, Baauw and Markopoulos (2004) determined that posttask interviews alone revealed fewer usability problems than think alouds; however, when combined with data from observations, which is standard practice, there was no significant difference between the problems that the two methods revealed.

Codiscovery. Codiscovery exploration is a usability method that is used to understand users' experiences and perceptions of new product designs, especially those that may be unfamiliar. In codiscovery sessions, two users who know one another work together to perform a set of tasks using the product. The goal of using two acquainted users is to encourage them to talk about the problems they encounter and their perceptions of the product in the natural course of collaborating on a task instead of relying on a single user's verbal performance for an experimenter (Kemp & van Gelderen, 1996). First, the two users are asked to figure out what a product does and compare it other products they know about. Next, they are asked to collaborate on a set of specific tasks using the product. Finally, a discussion period allows designers to ask about observed problems and behaviors; in addition, participants can ask questions about the design and the intended purpose of the product. Van Kesteren et al. (2003) found that using co-discovery with kids ages six and seven can be difficult because they often attempt to complete tasks individually. Even when seated next to one another, two children may not interact at all, resulting in very few comments about the product being tested. When compared with traditional usability tests, think alouds, posttask interviews, peer tutoring and active intervention, codiscovery was found to elicit the fewest comments from kids (van Kesteren et al., 2003).

Peer tutoring. Peer tutoring is a method for usability testing that was developed to capitalize on the ways that children interact with one another in natural, playful settings. When children play together, they regularly teach one another games and invent rules of play. Hoysniemi et al. (2003) explained, "One definition of the usability of a children's software application is that a child is able and willing to teach other children how to use it" (p. 209). Instead of relying on task completion in a lab, peer tutoring is an approach to usability testing that allows kids to engage in exploratory and playful interactions in a naturalistic setting. Peer tutoring involves first helping one or more kids to develop expertise using a piece of software and then asking them to teach other kids how to use it. By observing, recording, and analyzing interactions between tutors and tutees, it is possible to identify usability problems in software as the kids attempt to teach it to one another. Hoysniemi et al. (2003) pointed out that, although it can be useful, the peer-tutoring approach requires time, training, and careful implementation to be effective.

GENRES OF TECHNOLOGY FOR KIDS

Technology for kids falls into two broad categories: education and entertainment. When game companies try to mix these genres, they may use the term *edutainment.* New products for kids increasingly include specialized hardware as well as software.

Entertainment

Designers of games and other entertainment software rarely write about how they accomplish their jobs. Talks are presented each year at the Game Developer's Conference (http://www.gdconf.com), and some informal reflections are gathered as conference proceedings. Attending the conference is recommended for people who wish to learn more about current issues in game design. *Game Developer* magazine is the leading publication with reflective articles on the game design process.

Game designers are usually gamers themselves and often end up simply designing games that they themselves would like to play. This simple design technique is easy and requires little if any background research with users. Because most game designers have traditionally been male, this approach allowed them to appeal quite effectively to a core gaming audience: young men and teenage boys. However, female designers are becoming more common on design teams and gaming companies are increasingly recognizing that people outside the typical gamer stereotype represent a large potential market for their products. Designing for teenagers is relatively easy. As we have seen, designing for very young children presents substantial challenges. The younger your target audience, the more they should be tightly connected to every stage of the design process.

Brenda Laurel pioneered the use of careful design methods for nontraditional game audiences in her work with the company Purple Moon in the mid-1990s. Laurel aimed to develop games that appeal to preteen girls both to tap this market segment and to give girls an opportunity to become fluent with technology. Many people believe that use of computer game leads to skills that later give kids advantages at school and work. Through extensive interviews with girls in their target age range, Purple Moon was able to create successful characters and game designs. However, the process was so time consuming and expensive that the company failed to achieve profitability fast enough to please its investors. The company closed in 1999, and its characters and games were sold to Mattel. Purple Moon perhaps did more research than was strictly necessary, particularly because their area was so new. The broader lesson is that the game industry typically does not budget for needs analysis and iterative design early in the design process. Playtesting and quality assurance typically take place relatively late in the design cycle. Designers contemplating incorporating research early in their design process must consider the financial costs.

Oosterholt et al. (1996) described several design constraints that are specific to the design of products for children. First, they suggested that trying to be fashionable might result in products that kids quickly perceived as outdated. They also pointed out that fun is just as important to measure as usability, that measurements of fun should be shared with development teams, and moreover, that the product should grow with users over time and continue to be fun long after kids have learned to use it.

Game designer Carolyn Miller (1998) highlighted seven mistakes (or "kisses of death") commonly made by people trying to design games for kids:

"Death kiss #1: Kids love anything sweet"
Miller (1998) wrote, "Sweetness is an adult concept of what kids should enjoy." Only very young children will tolerate it. Humor and good character development are important ingredients. Do not be afraid to use off-color humor or to make something scary.

"Death kiss #2: Give 'em what's good for 'em"
Miller advised, "Don't preach, don't lecture, and don't talk down—nothing turns kids off faster."

"Death kiss #3: You just gotta amuse 'em"
Miller wrote, "Don't assume that just because they are little, they aren't able to consume serious themes."

"Death kiss #4: Always play it safe!"
Adult games often rely on violence to maintain dramatic tension. Since you probably will not want to include this in your game for kids, you will need to find other ways to maintain dramatic tension. Do not let your game become bland.

"Death kiss #5: All kids are created equal"
Target a specific age group, and take into consideration humor, vocabulary, skill level, and interests. If you try to design for everyone, your game may appeal to no one.

"Death kiss #6: Explain everything"
In an eagerness to be clear, some people overexplain things to kids. Kids are good at figuring things out. Use as few words as possible, and make sure to use spoken and visual communication as much as possible.

"Death kiss #7: Be sure your characters are wholesome!"
Miller warned that if every character is wholesome, the results are predictable and boring. Characters need flaws to have depth. Miller identified a number of common pitfalls in assembling groups of characters. It is not a good idea to take a "white bread" approach, in which everyone is White and middle class. On the other end of the spectrum, it is also undesirable to take a "lifesaver approach" with one character for each ethnicity. Finally, you also need to avoid an "off-the-shelf" approach, in which each character represents a stereotype: "You've got your beefy kid with bad teeth; he's the bully. You've got the little kids with glasses; he's the smart one." Create original characters that have depth and have flaws that they can struggle to overcome (Miller, 1998).

Education

To design educational software, we must expand the concept of user-centered design (UCD) to one of learner-centered design (LCD; Soloway, Guzdial, & Hay, 1994). There are several added steps in the process:

- Needs analysis
  - For learners
  - For teachers
- Select pedagogy
- Select media/technology
- Prototype
  - Core application
  - Supporting curricula
  - Assessment strategies
- Formative evaluation
  - Usability
  - Learning outcomes
- Iterative design
- Summative evaluation
  - Usability
  - Learning outcomes

In our initial needs analysis, for software to be used in a school setting, we need to understand not just learners but also teachers. Teachers have heavy demands on their time and are held accountable for their performances in ways that vary between districts and between election years.

Once we understand our learners and teachers needs, we need to select an appropriate pedagogy—an approach to teaching and learning. For example, behaviorism views learning as a process of stimulus and reinforcement (Skinner, 1968). Constructivism sees learning as a process of active construction of knowledge through experience. A social-constructivist perspective emphasizes learning as a social process (Newman, Griffin, & Cole, 1989). (A full review of approaches to pedagogy is beyond the scope of this chapter.)

Next, we are ready to select the media we will be working with, matching their affordances to our learning objectives and pedagogical approach. Once the prototyping process has begun, we need to develop not just software or hardware, but (for applications to be used in schools) also supporting curricular materials and assessment strategies.

Assessment should not be confused with evaluation. The goal of assessment is to judge an individual student's performance. The goal of evaluation is to understand to what extent our learning technology design is successful. An approach to assessing student achievement is an essential component of any school-based learning technology. For both school and free time use, we need to design feedback mechanisms so that learners can be aware of their progress. It is also important to note whether learners find the environment motivating. Does it appeal to all learners or to specific gender, learning style, or interest groups?

As in any HCI research, educational technology designers use formative evaluation to understand informally what needs improvement in their learning environments, and guide the process of iterative design. Formative evaluation must pay attention first to usability, and second to learning outcomes. If students cannot use the learning hardware or software, they certainly will not learn through its use. Once it is clear that usability has met a minimum threshold, designers then need to evaluate whether learning outcomes are being met. After formative evaluation and iterative design are complete, a final summative evaluation serves to document the effectiveness of the design and justify its use by learners and teachers. Summative evaluation must similarly pay attention to both usability and learning outcomes.

A variety of quantitative and qualitative techniques are commonly used for evaluation of learning outcomes (Gay & Airasian, 2000). Most researchers use a complementary set of both quantitative and qualitative approaches. Demonstrating educational value is challenging, and research methods are an ongoing subject of research.

This represents an idealized learner-centered design process. Just as many software design projects do not in reality follow a comprehensive UCD process, many educational technology projects do not follow a full LCD process. LCD is generally substantially more time consuming than UCD. While in some cases it may be possible to collect valid usability data in a single session, learning typically takes place over longer periods. To get meaningful data, most classroom trials take place over weeks or months. Furthermore, classroom research needs to fit into the school year at the proper time. If you are using Biologica (Hickey, Kindfield, Horwitz, & Christie, 2000) to teach about genetics, you need to wait until it is time to cover genetics that school year. You may have only one or two chances per year to test your educational technology. It frequently takes many years to complete the LCD process. In the research community, one team may study and evolve one piece of educational technology over many years. In a commercial setting, educational products need to get to market rapidly, and this formal design process is rarely used.

Genres of Educational Technology

Taylor (1980) divided educational technology into three genres:

- Computer as tutor
- Computer as tool
- Computer as tutee

Suppose that we are learning about acid rain. If the computer is serving as tutor, it might present information about acid rain and ask the child questions to verify the material was understood. If the computer is a tool, the child might collect data about local acid rain and input that data into an ecological model to analyze its significance. If the computer is a tutee, the child might program his or her own ecological model of acid rain.

With the advent of the Internet, we must add a fourth genre:

- Computer-supported collaborative learning (CSCL)

In a CSCL study of acid rain, kids from around the country might collect local acid rain data, enter it into a shared database, analyze the aggregate data, and talk online with adult scientists who study acid rain. This is in fact the case in the NGS-TERC Acid Rain Project (Tinker, 1993). See Table 40.1 for an overview of genres of children's software.

Computer as Tutor

In most off the shelf educational products, the computer acts as tutor. Children are presented with information and then quizzed on their knowledge. This approach to education is grounded in behaviorism (Skinner, 1968). It is often referred to as "drill and practice" or "computer-aided instruction" (CAI).

TABLE 40.1. Genres of Children's Software

Genre	Description
Entertainment	Games created solely for fun and pleasure.
Educational	Software created to help children learn about a topic using some type of pedagogy—an approach to teaching and learning.
Computer as Tutor	Often referred to as "drill and practice" or "computer-aided instruction" (CAI), this approach is grounded in behaviorism. Children are presented with information and then quizzed on their knowledge.
Computer as Tool	The learner directs the learning process, rather than being directed by the computer. This approach is grounded in constructivism, which sees learning as an active process of constructing knowledge through experience.
Computer as Tutee	Typically, the learner uses construction kits to help reflect upon what he or she learned through the process of creation. This approach is grounded in constructivism and constructionism.
Computer-supported Collaborative learning (CSCL)	Children use the Internet to learn from and communicate with knowledgeable members of the adult community. Children can also become involved in educational online communities with children from different geographical regions. This approach is grounded in social constructivism.
Edutainment	A mix of the entertainment and educational genres.

The computer tracks student progress and repeats exercises as necessary.

Researchers with a background in artificial intelligence have extended the drill and practice approach to create intelligent tutoring systems. Such systems try to model what the user knows and tailor the problems presented to an individual's needs. Many systems explicitly look for typical mistakes and provide specially prepared corrective feedback. For example, suppose a child adds 17 and 18 and gets an answer of 25 instead of 35. The system might infer that the child needs help learning to carry from the ones to the tens column and present a lesson on that topic. One challenge in the design of intelligent tutors is in accurately modeling what the student knows and what their errors might mean.

Byrne, Anderson, Douglass, and Matessa (1999) experimented with using eye tracking to improve the performance of intelligent tutors. Using an eye tracker, the system can tell whether the student has paid attention to all elements necessary to solve the problem. In early trials with the eye tracker, they found that some of the helpful hints the system was providing to the user were never actually read by most students. This helped guide their design process. They previously focused on how to improve the quality of hints provided; however, that is irrelevant if the hints are not even being read (Byrne et al., 1999).

An interesting variation on the traditional 'computer as tutor' paradigm for very young children is the Actimates line interactive plush toys. Actimates Barney and other characters lead children in simple games with educational value, like counting exercises. The tutor is animated and anthropomorphized. The embodied form lets young children use the skills they have in interacting with people to learn to interact with the system, enhancing both motivation and ease of use (Strommen, 1998; Strommen & Alexander, 1999).

Computer as Tool

When the computer is used as a tool, agency shifts from the computer to the learner. The learner is directing the process, rather than being directed. This approach is preferred by constructivist pedagogy, which sees learning as an active process of constructing knowledge through experience. The popular drawing program Kid Pix is an excellent example of a tool customized for kids' interests and needs. Winograd (1996) commented that Kid Pix's designer Craig Hickman "made a fundamental shift when he recognized that the essential functionality of the program lay not in the drawings that it produced, but in the experience for the children as they used it" (p. 60). For example, Kid Pix provides several different ways to erase the screen—including having your drawing explode or be sucked down a drain.

Simulation programs let learners try out different possibilities that would be difficult or impossible in real life. For example, Biologica (an early version was called "Genscope") allows students to learn about genetics by experimenting with breeding cartoon dragons with different inherited characteristics like whether they breathe fire or have horns (Hickey et al., 2000). Model-it lets students try out different hypotheses about water pollution and other environmental factors in a simulated ecosystem (Soloway et al., 1996).

The goal of such programs is to engage students in scientific thinking. The challenge in their designs is how to get students to think systematically and not to simply try out options at random. Programs like Model-It provide the student with scaffolding. Initially, students are given lots of support and guidance. As their knowledge evolves, the scaffolding is faded, allowing the learner to work more independently (Guzdial, 1994; Soloway et al., 1994).

Computer as Tutee

Papert (1992) commented that much computer-aided instruction is "using the computer to program the child" (p. 163). Instead, he argued that the child should learn to program the computer and through this process gain access to new ways of thinking and understanding the world. Early research argued that programming would improve children's general cognitive skills, but empirical trials produced mixed results (Clements, 1986; Clements & Gullo, 1984; Pea, 1984). Some researchers argued that the methods of these studies are fundamentally flawed, because the complexity of human experience cannot be reduced to pretests and posttests (Papert, 1987). The counterargument is that researchers arguing that technology has a transformative power need to back up their claims with evidence of some form, whether quantitative or qualitative (Pea, 1987; Walker, 1987). More recently, the debate has shifted to the topic of technological fluency. As technology increasingly surrounds our everyday lives, the ability to use it effectively as a tool becomes important for children's successes in school and later in the workplace (Resnick & Rusk, 1996).

In the late 1960s, Feurzeig (1996) and colleagues at BBN invented Logo, the first programming language for kids. Papert (1980) extended Logo to include turtle graphics, in which kids learn geometric concepts by moving a turtle around the screen. A variety of programming languages for kids have been developed over subsequent years, including Starlogo (Resnick, 1994), Boxer (diSessa & Abelson, 1986), Stagecast (Cypher & Smith, 1995), Agentsheets (Repenning & Fahlen, 1993), MOOSE (Bruckman, 1997), and Squeak (Guzdial & Rose, 2001). Lego Mindstorms (originally "Lego/Logo") is a programmable construction kit with physical as well as software components (Martin & Resnick, 1993). Another programmable tool bridging the gap between physical constructions and representations on the screen is Hypergami, a computer-aided design tool for origami developed at the University of Colorado at Boulder. Students working with Hypergami learn about both geometry and art (Eisenberg, Nishioka, & Schreiner, 1997).

In most design tools, the goal is to facilitate the creation of a product. In educational construction kits, the goal instead is what is learned through the process of creation. So what makes a good construction kit? In an *Interactions* article entitled "Pianos, Not Stereos: Creating Computational Construction Kits," Resnick, Bruckman, and Martin (1996) discussed the art of designing construction kits for learning (constructional design):

> The concept of learning-by-doing has been around for a long time. But the literature on the subject tends to describe specific activities and gives little attention to the general principles governing what kinds of

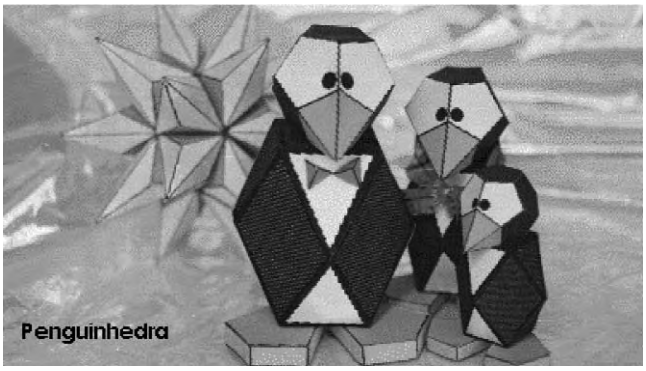


FIGURE 40.2. Penguins created using Hypergami.

"doing" are most conducive to learning. From our experiences, we have developed two general principles to guide the design of new construction kits and activities. These constructional-design principles involve two different types of "connections":

- *Personal connections.* Construction kits and activities should connect to users' interests, passions, and experiences. The point is not simply to make the activities more "motivating" (though that, of course, is important). When activities involve objects and actions that are familiar, users can leverage their previous knowledge, connecting new ideas to their pre-existing intuitions.
- *Epistemological connections.* Construction kits and activities should connect to important domains of knowledge—more significantly, encourage new ways of thinking (and even new ways of thinking about thinking). A well-designed construction kit makes certain ideas and ways of thinking particularly salient, so that users are likely to connect with those ideas in a very natural way, in the process of designing and creating.

Bruckman (2000) added a third design principle:

- *Situated support.* Support for learning should be from a source (either human or computational) with whom the learner has a positive personal relationship, ubiquitously available, richly connected to other sources of support, and richly connected to everyday activities.

Resnick and Silverman (2005) suggested several more design principles for creating construction kits for kids. Some, such as "iterate, iterate—then iterate again," are familiar mantras for HCI designers. Others may be less familiar:

- Low Floor and Wide Walls
  If a technology has a low floor, it means that it is easy for novices to begin using it. Wide Walls suggest a wide range of possible areas of design and exploration. Construction kits define "a place to explore, not a collection of specific activities."
- Make Powerful Ideas Salient—Not Forced
  When designing toward specific learning goals, construction kits should make these ideas visible and useful in design activities rather than imposing the ideas on students as a predetermined solution.
- Support Many Paths, Many Styles
  Kids approach problems in different ways, it is important to support a variety of design approaches in a construction kit.
- Make it as Simple as Possible—and Maybe Even Simpler
  Constraints can be the designer's best friend. Limited functionality sometimes wins out over designs that are more sophisticated because simplicity allows kids to find creative new ways to use a product.
- Choose Black Boxes Carefully
  This principle is related to the previous one; deciding when to reveal complexity and when to conceal it is a difficult question. Resnick and Silverman (2005) suggested that the simplest choice is often the best one.
- A Little Bit of Programming Goes a Long Way
  Because programming is the fundamental mode of construction with computers, designers of construction kits for kids often include some programming functionality. Focusing on powerful, simple commands that kids can do well is often the best way to support a diverse range of activities.
- Give People What They Want—Not What They Ask For
  Observations of kids can often tell designers more than their direct answers to questions. Kids may ask for unrealistic features or may not know themselves why they are having difficulty completing a task.
- Invent Things That You Would Want to Use Yourself
  Although they caution against overgeneralizing one's own personal likes and dislikes, Resnick and Silverman (2005) proposed that the most respectful approach to designing for kids is to create something that the designer herself finds enjoyable.

Computer-Supported Collaborative Learning (CSCL)

Most tools for learning have traditionally been designed for one child working at the computer alone. However, learning is generally recognized to be a social process (Newman et al., 1989). With the advent of the Internet came new opportunities for children to learn from one another and from knowledgeable members of the adult community. This field is called "Computer-Supported Collaborative Learning" (CSCL; Koschmann, 1996).

CSCL research can be divided into four categories:

1. Distance education
   Students attempt to use online environments in ways that emulate a traditional classroom.
2. Information retrieval
   Research projects in which students use the Internet to find information.
3. Information sharing

Students debate issues with one another. One of the first such tools was the Computer-Supported Intentional Learning Environment (CSILE), a networked discussion tool designed to help students engage in thoughtful debate as a community of scientists does (Scardamalia & Bereiter, 1994). They may also collect scientific data and share it with others online. In the "One Sky, Many Voices" project, students learn about extreme weather phenomena by sharing meteorological data they collect with other kids from around the world, and also by talking online with adult meteorologists (Songer, 1996). In the Palaver Tree Online

project, kids learn about history by talking online with older adults who lived through that period of history (Ellis & Bruckman, 2001). A key challenge in the design of information sharing environments is how to promote serious reflection on the part of students (Guzdial, 1994; Kolodner & Guzdial, 1996).

Technological Samba Schools

In *Mindstorms*, Papert (1980) had a vision of a "technological samba school." At samba schools in Brazil, a community of people of all ages gather together to prepare a presentation for carnival. "Members of the school range in age from children to grandparents and in ability from novice to professional. But they dance together and as they dance everyone is learning and teaching as well as dancing. Even the stars are there to learn their difficult parts" (Papert, 1980). People go to samba schools not just to work on their presentations, but also to socialize and be with one another. Learning is spontaneous, self-motivated, and richly connected to popular culture. Papert imagined a kind of technological samba school where people of all ages gather together to work on creative projects using computers. The Computer Clubhouse is an example of such a school in a face-to-face setting (Resnick & Rusk, 1996). MOOSE Crossing is an Internet-based example (Bruckman, 1998). A key challenge in the design of such environments is how to grapple with the problem of uneven achievement among participants. When kids are allowed to work or not work in a self-motivated fashion, typically some excel while others do little (Elliott, Bruckman, Edwards, & Jensen, 2000).

Child Safety Online

One challenge in the design of Internet-based environments for kids is the question of safety. The Internet does contain information that is sexually explicit, violent, and racist. Typically, such information does not appear unless one is looking for it; however, it is unusual but possible to stumble across it accidentally. Filtering software blocks access to useful information as well as harmful (Schneider, 1997). Furthermore, companies that make filtering software often fail to adequately describe how they determine what to block, and they may have unacknowledged political agendas that not all parents will agree with. Resolving this issue requires a delicate balance of the rights of parents, teachers, school districts, and children (Electronic Privacy Information Center, 2001). Another danger for kids online is the presence of sexual predators and others who wish to harm children. While such incidents are rare, it is important to teach kids not to give out personal information online such as their last names, addresses, or phone numbers. Kids who wish to meet online friends to friend face to face should do so by each bringing a parent and meeting in a well-populated public place like a fast-food restaurant. A useful practical guide "Child Safety on the Information Superhighway" is available from the Center for Missing and Exploited Children (http://www.missingkids.com). Educating kids, parents, and teachers about online safety issues is an important part of the design of any online software for kids.

CONCLUSION

To design for kids, we must have a model of what kids are and what we would like them to become. Adults were once kids. Many are parents. Some are teachers. We tend to think that we know kids—who they are, what they are interested in, and what they like. However, we do not have as much access to our former selves as many would like to believe. Furthermore, it is worth noting that our fundamental notions of childhood are in fact culturally constructed and change over time. Calvert (1992) wrote about the changing notion of childhood in America, and the impact it has had on artifacts designed for children and child rearing:

> In the two centuries following European settlement, the common perception in America of children changed profoundly, having first held to an exaggerated fear of their inborn deficiencies, then expecting considerable self-sufficiency, and then, after 1830, endowing young people with an almost celestial goodness. In each era, children's artifacts mediated between social expectations concerning the nature of childhood and the realities of child-rearing: before 1730, they pushed children rapidly beyond the perceived perils of infancy, and by the nineteenth century they protected and prolonged the perceived joys and innocence of childhood. (p. 8)

While Calvert (1992) reflected on the design of swaddling clothes and walking stools, the same role is played by new technologies for kids like programmable Legos and drill and practice arithmetic programs: These artifacts mediate between our social expectations of children and the reality of their lives. If you believe that children are unruly and benefit from strong discipline, then you are likely to design CAI. If you believe that children are creative and should not be stifled by adult discipline, then you might design an open-ended construction kit like Logo or Squeak. In designing for kids, it is crucial to become aware of one's own assumptions about the nature of childhood. Designers should be able to articulate their assumptions, and be ready to revise them based on empirical evidence.

References

Alborzi, H., Druin, A., Montemayor, J., Platner, M., Porteous, J., Sherman, L., et al. (2000). Designing StoryRooms: Interactive Storytelling Spaces for Children. *Proceedings of the Symposium on Designing interactive systems: Processes, practices, methods, and techniques,* Brooklyn, NY, 95–104.

Als, B. S., Jensen, J. J., & Skov, M. B. (2005). Comparison of think-aloud and constructive interaction in usability testing with children. *Proceedings of the 2005 Conference on Interaction Design and Children*, Boulder, CO, 9–16.

BBC. (1997). *Teletubbies Press Release*.

Baauw, E., & Markopoulos, P. (2004). A Comparison of think-aloud and post-task interview for usability testing with children. *Proceedings of the 2004 Conference on Interaction Design and Children*, College Park, MD, 115–116.

Bederson, B., Hollan, J., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local Tools: An Alternative to Tool Palettes. *Proceedings of the ACM Symposium on User Interface Software and Technology*, Seattle, WA, 169–170.

Berkovitz, J. (1994). Graphical Interfaces for Young Children in a Software-based Mathematics Curriculum. *Proceedings of the ACM Conference on Human Factors in Computing Systems: Celebrating Interdependence*, Boston, MA, 247–248.

Bernard, M.., Mills, M., Frank, T., & McKnown, J. (2001, Winter). Which fonts do children prefer to read online? *Software Usability Research Laboratory* (*SURL*). Retrieved March 16, 2007, from psychology .wichita.edu/surl/usabilitynews/w3/fontJR.htm

Bruckman, A. (1997). *MOOSE Crossing: Construction, community, and learning in a networked virtual world for kids.* Unpublished doctoral dissertation, MIT.

Bruckman, A. (1998). Community support for constructionist learning. *Computer Supported Cooperative Work, 7*, 47–86.

Bruckman, A. (2000). Situated support for learning: Storm's weekend with Rachael. *Journal of the Learning Sciences, 9*(3), 329–372.

Byrne, M. D., Anderson, J. R., Douglass, S., & Matessa, M. (1999). Eye tracking the visual search of click-down menus. *Proceedings of the ACM Conference on Human Factors in Computing Systems: the CHI is the limit*, Pittsburgh, PA, 402–409.

Calvert, K. (1992). *Children in the House: The Material Culture of Early Childhood, 1600–1900*. Boston: Northeastern University Press.

Clements, D. H. (1986). Effects of Logo and CAI Environments on Cognition and Creativity. *Journal of Educational Psychology, 78*(4), 309–318.

Clements, D. H., & Gullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology, 76*(6), 1051–1058.

Cypher, A., & Smith, D. C. (1995). End user programming of simulations. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Denver, CO, 27–34.

Davenport, A., & Wood, A. (1997). *TeleTubbies FAQ*. BBC Education. Retrieved date, from http://www.bbc.co.uk/cbeebies/teletubbies/grownups/faq.shtml

Dindler, C., Eriksson, E., Iversen, O. S., Ludvigsel, M., & Lykke-Olesen, A. (2005). Mission from Mars: A method for exploring user requirements for children in a narrative space. *Proceedings of the 2005 Conference on Interaction Design and Children*, Boulder, CO, 40–47.

diSessa, A. A., & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM, 29*(9), 859–868.

Donker, A., & Markopoulos, P. (2002). A comparison of think-aloud, questionnaires and interviews for testing usability with children. *Proceedings of HCI 2002*, location, 305–316.

Donker, A., & Reitsma, P. (2004). Usability testing with young children. *Proceedings of the Conference on Interaction Design and Children*, College Park, MD, 43–48.

Dourish, P (2001). *Where the action is: The foundations of embodied interaction*. Cambridge, MA: MIT Press.

Druin, A. (1999). Cooperative inquiry: developing new technologies for children with children. *Proceedings of the ACM Conference on Human Factors in Computing Systems: the CHI is the limit*, Pittsburgh, PA, 592–599.

Druin, A. (2002). The role of children in the design of new technology. *Behaviour & Information Technology, 21*(1), 1–25.

Druin, A., Bederson, B., Hourcade, J. P., Sherman, L., Revelle, G., Platner, M., et al. (2001). Designing a digital library for young children: An intergenerational partnership. *Proceedings of the Joint Conference on Digital Libraries*, Roanoke, VA, 398–405.

Eisenberg, M., Nishioka, A., & Schreiner, M. E. (1997). Helping users think in three dimensions: Steps toward incorporating spatial cognition in user modeling. *Proceedings of the International Conference on Intelligent User Interfaces*, Orlando, FL, 113–120.

Electronic Privacy Information Center. (2001). *Filters and Freedom 2.0: Free Speech Perspectives on Internet Content Control*. Washington DC: Author.

Elliott, J., Bruckman, A., Edwards, E., & Jensen, C. (2000). Uneven Achievement in a Constructionist Learning Environment. *Proceedings of the International Conference on the Learning Sciences*, Ann Arbor, MI, 157–163.

Ellis, J. B., & Bruckman, A. S. (2001). Designing palaver tree online: Supporting social roles in a community of oral history. *Proceedings of CHI: Conference on Human Factors in Computing Systems*, Seattle, WA, 474–481.

Erickson, T. (1990). Working with interface metaphors. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 65–73). Reading, MA: Addison Wesley Publishing Company Inc.

Ericsson, K. A., & Simon, H. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Gay, L. R., & Airasian, P. (2000). *Education Research: Competencies for Analysis and Application* (6th ed.). Upper Saddle River, New Jersey: Merrill.

Gilutz, S., & Nielsen, J. (2002). Usability of websites for children: 70 design guidelines based on usability studies with kids. Location: Nielsen Norman Group Report.

Goldman-Segall, R. (1996). Looking through layers: Reflecting upon digital video ethnography. *JCT: An Interdisciplinary Journal for Curriculum Studies, 13*(1), 23–29.

Guha, M. L., Druin, A., Chipman, G., Fails, J. A., Simms, S., & Farber, A. (2005). Working with young children as technology design partners. *Communications of the ACM, 48*(1), 39–42.

Guzdial, M. (1994). Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments, 4*(1), 1–44.

Guzdial, M., & Rose, K. (Eds.). (2001). *Squeak: Open Personal Computing and Multimedia*. Englewood, NJ: Prentice Hall.

Halgren, S., Fernandes, T., & Thomas, D. (1995). Amazing Animation™: Movie making for kids design briefing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, 519–525.

Hanna, L., Risden, K., & Alexander, K. (1997). Guidelines for usability testing with children. *Interactions, 4*(5), 9–14.

Hickey, D. T., Kindfield, A. C. H., Horwitz, P., & Christie, M. A. (2000). Integrating instruction, assessment, and evaluation in a technology-based genetics environment: The GenScope follow-up study. *Proceedings of the International Conference of the Learning Sciences*, Ann Arbor, MI, 6–13.

Holdaway, D. (1979). *The Foundations of Literacy*. New York: Ashton Scholastic.

Hourcade, J. P. (2002). *It's too small! Implications of children's developing motor skills on graphical user interfaces* (No. CS-TR-4425). University of Maryland Computer Science Department.

Hoysniemi, J., Hamalainen, P., & Turkki, L. (2003). Using peer tutoring in evaluating the usability of a physically interactive computer game with children. *Interacting with Computers, 15*, 203–225.

Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B. B., Druin, A., Plaisant, C., et al. (2003). Technology probes: Inspiring design for and with families. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL, 17–24.

Inkpen, K. (1997). Three important research agendas for educational multimedia: Learning, children and gender. *Proceedings of Educational Multimedia '97*, Calgary, AB, 521–526.

Inkpen, K. (2001). Drag-and-drop versus point-and-click: Mouse interaction styles for children. *ACM Transactions Computer-Human Interaction, 8*(1), 1–33.

Inkpen, K., Gribble, S., Booth, K. S., & Klawe, M. (1995). Give and take: Children collaborating on one computer. *Proceedings of the ACM*

Conference on Human Factors in Computing Systems, Denver, CO, 258–259.

Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of CHI '97*, 234–241.

Iversen, O. S. (2002). Designing with children: The video camera as an instrument of provocation. *Proceedings of the Conference on Interaction Design and Children*, Eindhoven, Netherlands, CD-ROM.

Jensen, J. J., & Skov, M. B. (2005). A review of research methods in children's technology design. *Proceedings of the Conference on Interaction Design and Children*, Boulder, CO, 80–87.

Joiner, R., Messer, D., Light, P., & Littleton, K. (1998). It is best to point for young children: A comparison of children's pointing and dragging. *Computers in Human Behavior, 14*(3), 513–529.

Jones, T. (1992). Recognition of animated icons by elementary-aged children. *Association for Learning Technology Journal, 1*(1), 40–46.

Kail, R. (1991). Developmental Changes in speed of processing during childhood and adolescence. *Psychological Bulletin, 109*, 490–501.

Kay, A. (1972). A personal computer for children of all ages. *Proceedings of the ACM National Conference*, Boston, MA.

Kemp, J. A. M., & van Gelderen, T. (1996). Codiscovery exploration: An informal method for the iterative design of consumer products. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 139–146). London: Taylor & Francis Ltd.

Kolodner, J., & Guzdial, M. (1996). Effects with and of CSCL: Tracking learning in a new paradigm. In T. Koschmann (Ed.), *CSCL: Theory and practice*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Koschmann, T. (Ed.). (1996). *CSCL: Theory and Practice*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Lamberty, K. K., & Kolodner, J. (2005). Camera talk: Making the camera a partial participant. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, OR, 839–848.

MacFarlane, S., Sim, G., & Horton, M. (2005). Assessing usability and fun in educational software. *Proceedings of the Conference on Interaction Design and Children*, Boulder, CO, 103–109.

Markopoulos, P., & Bekker, M. (2002). How to compare usability testing methods with children participants. *Proceedings of the Conference on Interaction Design and Children*, Eindhoven, Netherlands, 153–158.

Martin, F., & Resnick, M. (1993). LEGO/Logo and Electronic Bricks: Creating a scienceland for children. In D. L. Ferguson (Ed.), *Advanced Educational Technologies for Mathematics and Science* (pp. 61–90). Berlin Heidelberg: Springer-Verlag.

McElligott, J., & van Leeuwen, L. (2004). Designing sound tools and toys for blind and visually impaired children. *Proceedings of the Conference on Interaction Design and Children*, College Park, MD, 65–72.

Miller, C. (1998). Designing for kids: Infusions of life, kisses of death. *Paper presented a the Game Developers Conference*, Longbeach, CA.

Miller, L. T., & Vernon, P. A. (1997). Developmental changes in speed of information processing in young children. *Developmental Psychology, 33*(4), 549–554.

Newman, D., Griffin, P., & Cole, M. (1989). *The Construction Zone: Working for Cognitive Change in School*. Cambridge, England: Cambridge University Press.

Nielsen, J. (1993). *Usability Engineering*. London: Academic Press.

Nix, D., Fairweather, P., & Adams, B. (1998). Speech recognition, children, and reading. *Proceedings of the SIGCHI Conference on Human Factors in Computings Systems,* Los Angeles, CA, 245–246.

O'Hare, E. A., & McTear, M. F. (1999). Speech recognition in the secondary school classroom: an exploratory study. *Computers & Education, 3*(8), 27–45.

O'Malley, C., & Fraser, D., (2004). *Literature Review in Learning with Tangible Technologies*. NESTA Futurelab Report.

Oosterholt, R., Kusano, M., & Vries, G. D. (1996). Interaction design and human factors support in the development of a personal communicator for children. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, Canada, 450–457.

Papert, S. (1972). *On making a theorum for a child*. Paper presented at the ACM National Conference, Boston, MA.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Papert, S. (1987). Computer criticism vs. technocentric thinking. *Educational Researcher, 16*(1), 22–30.

Papert, S. (1992). *The Children's Machine.* New York: Basic Books.

Pea, R. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology, 2*(2), 137–168.

Pea, R. (1987). The aims of software criticism: Reply to Professor Papert. *Educational Researcher, 16*, 4–8.

Piaget, J. (1970). *Science of Education and the Psychology of the Child*. New York: Orion Press.

Price, S., Rogers, Y., Scaife, M., Stanton, D., & Neale, H. (2003). Using 'tangibles' to promote novel forms of playful learning. *Interacting with Computers, 15*(2), 169–185.

Repenning, A., & Fahlen, L. E. (1993). Agentsheets: A tool for building domain-oriented visual programming environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Amsterdam, Netherlands, 142–143.

Resnick, M. (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.

Resnick, M., Bruckman, A., & Martin, F. (1996). Pianos not stereos: Creating computational construction kits. *Interactions, 3*(5), 40–50.

Resnick, M., & Rusk, N. (1996). The Computer Clubhouse: Preparing for life in a digital world. *IBM Systems Journal, 35*(3–4), 431–440.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceedings of the Conference on Interaction Design and Children*, Boulder, CO, 117–122.

Rubin, J. (1994). *Handbook of Usability Testing*. New York: John Wiley and Sons, Inc.

Scaife, M., & Rogers, Y. (1999). Kids as informants: Telling us what we didn't know or confirming what we knew already? In A. Druin (Ed.), *The design of children's technology* (pp. 27–50). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Scaife, M., Rogers, Y., Aldrich, F., & Davies, M. (1997). Designing for or designing with? Informant design for interactive learning environments. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, Atlanta, GA, 343–350.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *The Journal of the Learning Sciences, 3*(3), 265–283.

Schneider, K. G. (1996). Children and information visualization technologies. *Interactions, 3*(5), 68–73.

Schneider, K. G. (1997). *The Internet Filter Assessment Project* (*TIFAP*). Retrieved March 16, 2007, from http://www.bluehighways.com/tifap/learn.htm

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory design: Principles and practices*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers.

Skinner, B. F. (1968). *The technology of teaching*. New York: Appleton-Century-Crofts.

Soloway, E., Guzdial, M., & Hay, K. E. (1994). Learner-centered design: The challenge for HCI in the 21st century. *Interactions, 1*(1), 36–48.

Soloway, E., Jackson, S. L., Klein, J., Quintana, C., Reed, J., Spitulnik, J., et al. (1996). Learning theory in practice: Case studies of learner-centered design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, Canada, 189–196.

Songer, N. B. (1996). Exploring learning opportunities in coordinated network-enhanced classrooms: A case of kids as global scientists. *The Journal of the Learning Sciences, 5*(4), 297–327.

Stewart, J., Raybourn, E. M., Bederson, B., & Druin, A. (1998). When two hands are better than one: enhancing collaboration using single display groupware. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Los Angeles, CA, 287–288.

Strommen, E. (1994). Children's use of mouse-based interfaces to control virtual travel. *Proceedings of the ACM Conference on Human Factors in Computing Systems: Celebrating Interdependence*, Boston, MA, 229.

Strommen, E. (1998). When the Interface is a Talking Dinosaur: Learning Across Media with ActiMates Barney. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Los Angeles, CA, 288–295.

Strommen, E., & Alexander, K. (1999). Emotional Interfaces for Interactive Aardvarks: Designing Affect into Social Interfaces for Children. *Proceedings of the ACM Conference on Human Factors in Computing Systems: the CHI is the limit*, Pittsburgh, PA, 528–535.

Taylor, R. P. (Ed.). (1980). *The computer in the school, tutor, tool, tutee.* New York: Teachers College Press.

Thomas, J. R. (1980). Acquisition of motor skills: Information processing differences between children and adults. *Research Quarterly for Exercise and Sport, 51*(1), 158–173.

Tinker, R. (1993). *Thinking about science*. Concord, MA: The Concord Consortium.

van Kesteren, I. E., Bekker, M. M., Vermeeren, A. P. O. S., & Lloyd, P. A. (2003). Assessing usability evaluation methods on their effectiveness to elicit verbal comment from children subjects. *Proceedings of the Conference on Interaction Design and Children*, Preston, UK, 41–49.

Walker, D. F. (1987). Logo needs research: A response to Professor Papert's paper. *Educational Researcher*, 9–11.

Winograd, T. (1996). Profile: Kid Pix. In T. Winograd (Ed.), *Bringing design to software* (pp. 58–61). New York: ACM Press.

◆41◆

INFORMATION TECHNOLOGY FOR COGNITIVE SUPPORT

A. F. Newell, A. Carmichael, P. Gregor, N. Alm, and A. Waller
University of Dundee

Introduction **812**
Cognitive Impairment **812**
Intelligence Quotient **813**
Interface Design to Support People With Cognitive Impairment **814**
The Effects of Attention 814
Memory Loss and Dementia 815
Visuo-Spatial, Iconic, and Verbal Abilities 815
Augmentative and Alternative Communication 816
Cognitive Prostheses 816
Software as a Cognitive Scaffolding and a Prompt for Communication 817
Help for Aphasia 819
Support for Nonliterate Users 819
Support for Dementia 820
Systems to Support People With Dyslexia **821**
Some Common Problems of Dyslexia 822
Computer Aid for the Problems of Dyslexia 822
An Experimental Text Reader for Dyslexics 822
Research Methodologies **823**
The Disabling Environment 825
The Inclusion of Users With Disabilities Within Research Groups 826
User Sensitive Inclusive Design 826
Design for Dynamic Diversity 826
A story-telling metaphor 826
The use of theater 826
Conclusion **827**
References **827**

INTRODUCTION

Well-designed communication and information technology systems have great, unrealized potential to enhance the quality of life and independence for those with cognitive dysfunction, including elderly people, by

- Allowing them to retain a high level of independence and control over their lives
- Providing appropriate levels of monitoring and supervision of at-risk people, without violating privacy
- Keeping people intellectually and physically active
- Providing communication methods to reduce social isolation and foster social inclusion

Using technology to augment human cognitive capacity is not a new idea, taking the term *technology* in its widest sense, to include tools and techniques which humans have developed to help them improve upon their limitations. The first cognitive function to be augmented was probably memory. Mnemonic methods help with this and are still in wide use today. One of the most common (e.g., the method of loci) is to link what is to be remembered with a well-established memory structure that is easy to recall, such as the layout of a familiar city, or a narrative that has already been memorized. The introduction of written language extended people's cognitive abilities by allowing memories to be recorded externally. Although at the time some feared this backwards step would allow people's memory powers to wither, it has since become apparent that the overriding effect has been to free cognitive abilities that can further develop on the basis of the external support. The development of information processing technologies has continued this trend of extending the potential of most people's cognitive abilities. Thus, there should be similar potential for computer technology to extend the cognitive abilities of those with some form of impairment or other limitation.

The increasing power and decreasing size of computer technology, along with its capacity to provide communication as well as computation and storage, offers the possibility of quite sophisticated help for cognitive impairments, if we can develop the appropriate software. Computers have the potential to act as a kind of scaffolding for cognitive tasks, taking over functions that have been affected by illness, accident, or aging. They could also provide prompts for daily living, if they were able to track successfully through the user's sequence of tasks and actions.

Particular strengths of computers as assistants for people with cognitive impairments include being consistent, tireless, and not becoming emotionally involved. In addition, multimedia and multimodal systems can provide a very rich interaction, which may be particularly advantageous for users with cognitive dysfunction. For example, such systems have great potential in addressing the problems of memory loss and the related difficulties presented by dementia. In addition, communication systems that use synthetic speech, predictive programs that can facilitate writing, and a range of nonlinguistic methods of communication can be used by those with speech and language dysfunction caused by cognitive impairments. A great deal of work will be required in order to realize this potential both in uncovering new knowledge about how cognition works and in developing assistive systems based on this knowledge.

Because of the wide range of skills and knowledge needed to understand the problems faced by people with cognitive impairment, it is essential that research work in this field should be multidisciplinary, including psychologists, members of the health and therapeutic professions, and engineers. It is also vital to involve potential users of the technology as partners in the research at all stages of development.

In addition, it is essential that interface designers be much more aware of the range of abilities that exist in any population of users. For example, access to the Internet for disabled people is often thought to be synonymous with access for blind people, but in fact, blind people form only a small percentage of the disabled population. Ogozalec (1997) pointed out that, if current trends continue in the United States, by 2030, one fifth of the population will be over 65 years of age and commented, "It is difficult to categorize and draw conclusions about 'the elderly', since they comprise such a diverse and heterogeneous population" (p. 65). This diversity, particularly of cognitive function, ought to be taken into account if we are to make software and the Internet available to as large a percentage of the population as possible.

We will describe the major types of cognitive impairment and include within this discussion the effects of aging. We will then illustrate the development of systems to support people with a variety of cognitive impairment with specific projects in which we have been involved. They will not cover all aspects of cognitive impairment, but they do illustrate a methodology and an approach to developing assistive technology, which may have a wider relevance.

In the concluding part of this chapter, we will address the development of methodologies that we believe will be valuable for designers of systems to support people with cognitive impairment and to assist designers of general systems to take into account the needs of people with cognitive impairments.

COGNITIVE IMPAIRMENT

The use of the term *cognitive impairment* implies that two categories of human cognitive systems exist—impaired and unimpaired. However, this is not the case, although it can reasonably be stated that there are normal or average cognitive systems. The vast majority of experimental cognitive-psychology literature relates to this normal system. In many contexts, this level of explanation is suitable for indicating of what most people are capable. It should always be borne in mind, however, that in real-world situations there is no marked distinction between that which is normal and that which is not. In other words, everyone has some limits to their cognitive abilities. Some have a highly specific impairment, some have more diffuse problems, and some experience interrelated constellations of impairments. In addition, the cognitive abilities of any one human being will change over longer and shorter periods. Aging can have substantial effects on cognitive ability, which is particularly marked in some age related conditions, such as dementia.

For ease of exposition, the forms of cognitive impairment identified and described below will in the main refer to general categories. It should be noted, however, that all these categories lie somewhere on a continuum and, while they are delineated on the basis of educational and clinical or medical criteria, such cut off points are relatively arbitrary in the context of the wide variability of cognitive ability across the population. It is also worth noting that within the context of normal cognitive systems, there is significant diversity among people in regard to differential preferences for types of material and ways of approaching and processing information. For example, some people may be considered primarily verbal and tend to excel in language-based tasks, relative to those considered visuo/spatial (e.g., Lohman, 2000). Thus, many of the types of impairment addressed below can be construed to an important extent as the extremities of normal diversity.

INTELLIGENCE QUOTIENT

The most widely known dimension of general cognitive ability is probably intelligence. Scientific investigation of this dimension has a controversial past, and many aspects of this are beyond the scope of the present chapter (for a more comprehensive account, see Gould, 1997). One underlying reason for such controversy is that the word *intelligence* has a rather nebulous definition. In day-to-day usage, this is rarely problematic, but the differences between scientific and lay definitions can cause misunderstanding (e.g., Sternberg, 2000). Such misunderstanding can lead to controversy as most definitions of intelligence include connotations that are considered socially important and, thus, can often be highly emotive. Despite these difficulties, the investigation of intelligence has provided many insights into a wide range of more particular cognitive abilities, and has developed methods for quantifying general intellectual ability such as the various forms of IQ (intelligence quotient) tests. Again, controversy has surrounded the use of these tests over the years (see Gould, 1997; Kaufman, 2000), but such tests have been widely used and are accepted as a general benchmark of a person's intellectual capability.

An IQ score of 100 is, by definition, normal with about 50% of the population scoring above and 50% scoring below, but it should be noted that elderly people are not generally included in the standardization of these scores. Approximately 50% of the population is considered within the bounds of normality and deviate either side of 100 by no more than 10 points. The nonnormal 50% are distributed evenly above and below this band. Thus about a quarter of the (nonelderly) population fall below the level of what is considered normal. Although the terminology varies across cultures and over time, around 20% of the population have IQ scores between 75 and 90 and would generally be classified as slow learners. The final 5% will generally have very special needs that overall are best addressed on an individual basis (Kaluger & Kolson, 1987). Further to this and as an example of the emotive connotations associated with the issue of 'intelligence', it is worth noting that the first official classification scheme (see Detterman, Gabriel, & Ruthsatz, 2000) associated with IQ tests further broke down this latter 5%. These classifications were moron (IQ 50–75), imbecile (IQ 25–50), and idiot (IQ < 25), terms which today would be considered wholly unacceptable as a description of anyone with a cognitive impairment.

An IQ score reflects a person's intellectual ability as a whole. A low score may be due to the whole system functioning at a suboptimal level, but a similar result can also be due to one or more component abilities being impaired. There are many tests of IQ, some of which give an indication of this while others do not. Some IQ tests are explicitly broken down into subtests that reflect the relative levels of ability in the component cognitive abilities, such as the verbal and visuo/spatial abilities mentioned above. The more common forms of cognitive impairment are described below in the context of a brief overview of the cognitive system.

For any information in the outside world to enter the cognitive system, it must first be detected and transmitted by the sensory apparatus. In an important sense, this is not simply the start of the process because aspects of attention will influence what is and is not detected/transmitted, and, to a certain extent, how. Basic perceptual processing creates a sensory specific representation of the stimulus event. Streams of such stimulus events are summated into meaningful cognitive entities (e.g., strokes on a page recognized as letters and numerals are summated into a name and telephone number). These will then be either passed immediately to short-term memory (STM), further processed by working memory (WM), or rehearsed for maintenance in STM or for encoding into longer term storage. For example, rehearsal could refer to rote rehearsal of a telephone number between reading and dialing it or to more elaborate processing to associate it with relevant extant memories to improve the chance of subsequent recall (e.g. method of loci, mentioned above).

Output from the cognitive system will generally be initiated in response to some form of external stimulus, or probe, by accessing extant memories relevant to the probe using executive processes to organize them in a task relevant way and then producing a response. Output of this kind has been most commonly studied with the use of memory tests. This minimizes the influence of intellectual processing (problem solving, etc.) per se, and emphasizes the registration, rehearsal and encoding of information, the effects of decay, interference and other forms of forgetting, and the effectiveness of different cues (probes) in eliciting specific memories (e.g., recall vs. recognition).

Virtually all aspects of the processing outlined above are shaped by attention, and it is important to note that, regardless of impairment, while we all have some control over attention, it can also be the case that attention can have some control over us. That is, we can utilize attention to focus on searching a list for a particular telephone number while ignoring the chatter of people around us. Having read the number, however, our attention can exert its own control if someone calls our name and asks if we have made that call yet. Despite our best efforts at rehearsal, it is likely that our attention will be grabbed by our name, and the ensuing question and this brief distraction can be enough to lose the information from temporary storage.

In general terms, mild to moderate global cognitive impairment will be associated with decrements in efficiency across most of the processing stages outlined above and in aspects of

the utilization of attention. The following will describe some of the main decrements in cognitive ability related to interacting with computer type systems.

INTERFACE DESIGN TO SUPPORT PEOPLE WITH COGNITIVE IMPAIRMENT

It is often important to develop special technology to provide support for people with various types of cognitive impairment, and some such projects are described later. It is also important, however, to address the challenge of providing access to more mainstream technology for people with cognitive impairments. When designing or specifying mainstream technology for such users, it is important to focus on their characteristics and to be fully aware of the range of cognitive diversity, even amongst those without clinical dysfunction. This is rarely mentioned in human interface design, where the cognitive diversity of the human race has not been the focus of much research. It is also important to consider the effects of age on cognitive function. As Worden, Walker, Bharat, and Hudson (1997) commented:

> It is known that, as people age, their cognitive perceptual and motor abilities decline with negative effects on their ability to perform many tasks. Computers can play an increasingly important role in helping older adults function well in society. Despite this little research has focused on computer use of older adults. (p. 266)

A key aspect of any intellectual task, in regard to interactive technology for people with mild or moderate global cognitive impairment, is speed (Salthouse, T. A., 1991). Whatever level of performance a person can achieve in any given situation will be made worse if the task must be done under externally imposed time constraints whether actual or simply inferred by the user. Thus, wherever possible, the design of any interaction should allow every step to be carried out at the user's own pace. This issue also raises the first distinction between older and younger people. A relatively greater proportion of the extra time needed by older people is due to age-related declines in their sensory systems, particularly in hearing and vision rather than cognitive impairment per se. For example, given comparable levels of cognitive impairment, an older person will need relatively more time to perceive the relevant stimulus before cognitive processes can be brought to bear on it. Thus, for many older people, the requirement for extra time can be reduced, though rarely removed, if care is taken to present text and other aspects of on-screen layout in a suitably clear way (e.g., Carmichael, A. R., 1999; Charness & Bosman, 1994). Attention to text layout can also benefit people with specific learning difficulties, such as those with dyslexia and people with limited literacy levels. Beyond this, clear text and presentation layout will always be worth considering carefully as such aspects have been found to benefit those who do not specifically need it, albeit to a less marked extent (Freudenthal, 1999; Pirkl, 1994).

Another key concept related to interface design for people with cognitive impairment is complexity and its avoidance. Complexity in interfaces can manifest itself in many different ways and at many different levels. A truly comprehensive coverage of this is beyond the scope of the present chapter, but some illustrative examples will be given to elucidate this idea.

The use of language in an interactive system should be given careful consideration, and the syntax and vocabulary should be kept as straightforward and commonplace as the context allows. This is particularly pertinent for any form of instruction. If the requirements of a particular stage of an interaction cannot be captured in a few simple concrete statements, then serious consideration should be given to redesigning the interaction itself. Similarly any on-screen display should be kept as uncluttered as practicable and wherever possible should present the user with only a single issue (menu, subject, decision, etc.) at any particular point in time. Similarly, but at a larger scale, progression through an interaction should be kept, again wherever practicable, as linear as possible. That is, the user should only need to consider one thing at a time. Any requirement to deal with different things in parallel will markedly increase the possibility of errors and general user dissatisfaction (Detterman et al., 2000; Salthouse, T., 1985).

Unfortunately, as the designers of a system will have a comprehensive understanding of the functions of that system, they are unlikely to assess issues of complexity from the users' point of view, particularly that of a novice user. In addition, prescriptive checklists for avoiding complexity will ultimately be inadequate, as the optimum approach will always depend of the specifics of the task the interactive system is intended to support (Carmichael, A. R., 1999). This is one of the main issues that highlights the importance of early and rigorous user involvement in the design of interactive systems. This is particularly important in the case of young designers developing interfaces for older users.

Research into cognitive changes in later life has indicated the heuristic value of the concepts of fluid and crystallized abilities (Horn & Cattell, 1967). In general, fluid abilities (novel problem solving) are those that decline with age, and crystallized abilities (existing world knowledge) are those that do not. Research that has looked at the very old (80 years and older) found that ultimately everything declines but that the crystallized abilities follow a markedly slower trajectory (Bäckman, Small, Wahlin, & Larsson, 2000). This is another distinction between younger and older people, as the former will tend to have much less accumulated knowledge. It is also worth noting that there are advantages and disadvantages on both sides of this distinction. Rabbitt (1993) has shown that, in many circumstances, relevant accumulated knowledge can ameliorate decline in fluid ability (e.g. well-learned strategies). In other circumstances, however, the opposite can be the case, wherein a well-learned, but essentially inappropriate, strategy can put a relatively greater burden on the associated fluid abilities.

The Effects of Attention

Many of the constraints imposed by cognitive impairment can be further shaped by decrements in various aspects of attention. One major aspect of this is generally referred to as "selective attention," which allows us to focus on salient aspects of a task and at the same time helps us actively to ignore irrelevant aspects. The efficiency of selective attention is markedly dimin-

ished in most forms of cognitive impairment. This factor further supports the recommendation to present the user with just one thing at a time, which will avoid the user erroneously devoting time and cognitive resources to processing irrelevant information. Similarly, if the nature of the interaction requires the user to attend to some critical information at a particular time/location, appropriately obvious highlighting should be employed to grab the users' attention. These issues become emphasized in situations where selective attention must be maintained over periods of more than just a few minutes.

Another aspect of attention known to be less efficient in cognitive impairment is referred to as "divided attention." In general, this refers to the ability to allocate cognitive resources appropriately when trying to do two or more distinct cognitive tasks or distinct portions of the same task at the same time. Many scenarios, where the user is required to do more than one thing at a time, can simply demand more cognitive resources than are available. However, declines in the efficiency of divided attention can mean that even if the tasks involved demand no more than the resources available, they may not be allocated appropriately. Generally, the interactive system should be designed to relieve the user of this kind of burden. It is difficult to be prescriptive about suitable solutions to this problem, as the appropriate approach will depend on the specifics of the interaction involved, but some general ideas may be of use. For example, the provision of some form of notepad function may be helpful for temporarily recording information for subsequent use, although great care is needed to ensure that the instantiation of such a function and its utilization does not put further cognitive load on the user. Another possibility would be the provision of an overview of the task, which could show or remind the user where they are and what they have and have not done.

Memory Loss and Dementia

Limitations in memory affect people with age- and nonage-related cognitive impairment. Thus, wherever practicable, interactive systems should be designed to take the burden of memory off the user, for example, by judicious use of prompts and reminders. Also, careful consideration of the steps in an interaction and the way they are presented to the user can help mitigate the most common problem of deficient short term and working memory. Even with the best design efforts, however, such problems are likely to make users with cognitive impairment relatively error prone. It is thus very important to ensure that the interactive system allows for error correction in an easy to use form. To ensure that the user spots such errors, the system should provide feedback regarding user actions and where appropriate elicit active confirmation from them.

Various forms of dementia can exaggerate the relatively mild effects of normal aging on the cognitive system. At the age of 60 years, about 1% of the population is diagnosed with dementia. This percentage approximately doubles for every subsequent 5-year age band (e.g., 4% at 70 years and 16% at 80 years; Bäckman et al., 2000). Of the elderly population with dementia, Alzheimer's disease accounts for about 60%, depending on the diagnostic criteria used and a further 25% is vascular dementia (e.g., related to circulatory problems). Most of the vascular dementias are referred to as "multi-infarct dementias" and tend to be caused by series of ministrokes, and thus, they tend to have more diffuse and less predictable effects on ability than a major stroke. The remaining 15% of dementias are made up of various relatively rare conditions (Bäckman et al., 2000).

Regardless of the various causes and effects, all forms of dementia involve damage to the brain, such damage being more or less widespread, affecting cortical and/or subcortical areas. In general, damage to the cortex results in cognitive/perceptual impairment, while damage to the subcortical areas is more related to physical impairment. There are, however, a number of well-known problems related to the diagnosis of dementia. Two of these are relevant here. The first involves the grey area between the worst effects of normal aging and the initial effects of pathological aging at the onset of dementia. The second is that the effects of depression in later life can closely mimic those of dementia. These additional complexities further expand the overall diversity of cognitive impairment in relation to human interface design, both in regard to the general level of ability and in the variation of that level over periods of days, weeks, months, and even years. The convolution of this situation is further added to by the effects of a relatively greater probability of ill health among older people. It is estimated that around 80% of those aged 65 and over have at least one chronic illness and that many will have more than one. In addition to the effects of health per se, there is also potential for cognitive ability to be affected by a variety of medication and by interactions between different medicines.

Despite the above, some systematic changes associated with extreme old age and dementia are relevant to human-interface design. In general, the first ability to deteriorate in dementia, particularly with Alzheimer's disease, is episodic as distinct from semantic memory. Episodic memory is memory for events, usually from the viewpoint of personal experience, rather than for facts. That is, remembering, "X is the capital of Y" is the product of semantic memory, whereas remembering when and where you were while you were reading "X is the capital of Y" is the product of episodic memory. This generalized decrement in episodic memory may be related to findings in normal aging research such as disproportionate decrements in source memory (e.g., specifically remembering where an item was rather than what it was) and to prospective memory (e.g., remembering to do something in the future). These changes have important implications for successful navigation in interactive systems. For example, keeping track of where you have just been, is often an important prompt to where you are going now.

Visuo-Spatial, Iconic, and Verbal Abilities

In addition to memory problems, there are, at least at the level of the population, marked deterioration of visuo-spatial and verbal abilities in older people. Decline in visuo-spatial abilities can cause difficulty with decoding layouts and utilizing any inherent organization. Related to this is a deterioration in iconic memory which, given the graphical nature of many interfaces, can be problematic in its own right. Particular difficulties have also been found in the comprehension of abstract and metaphorical phrases, with the tendency being to take them literally.

Such conditions can develop into more global aphasia (e.g., Broca's aphasia, related to the production of speech, and Wernicke's aphasia, related to comprehension). The general difficulty with recall of proper nouns, found in normal aging, can likely develop into more profound anomia. The depth of such problems may not be apparent to an outside observer as the ability to read aloud may be well preserved, although the content may not be properly understood

Another distinct form of global cognitive impairment is autism, including a set of rarer but related syndromes (Kaluger & Kolson, 1987). The precise causes of autism are not clearly understood. Briefly stated, it is a general neurological disorder that impacts the normal development of the brain particularly in relation to social interaction and communication skills. Its effects will usually become apparent within the first three years of life. People with autism typically have difficulties in verbal and nonverbal communication and social interactions. The disorder makes it hard for them to communicate with others and relate to the outside world, they will also tend to have relatively low IQ scores. Closely related to autism is Asperger's Syndrome. People with Asperger's experience similar social communication difficulties but will generally demonstrate a normal IQ. Further to this, there are several generally similar conditions, some of which have varying physical and behavioral elements associated with them. These come under the collective heading of pervasive developmental disorders and all tend to produce difficulties with communication. An important element of these social communication difficulties in the context of the present chapter is an inability to grasp the implications of metaphorical or idiomatic language. This is similar to that mentioned above for dementia but in autism tends to be more profound. However, some evidence shows that people with autism or Asperger's syndrome are more able to communicate with computers than with people, or with people via computers, rather than face to face, and thus properly designed computer systems may have potential for assisting such user groups. A variety of other impairments could also be ameliorated with suitable designed support for communication.

Augmentative and Alternative Communication

Most interfaces use traditional orthography (the written word). However, cognitive limitations can have a major impact on the ability to encode and decode traditional orthography. People with acquired cognitive impairments (e.g. aphasia resulting from a stroke or cerebral vascular accident; CVA) can experience varying degrees of difficulty with both expressive and receptive language. A person with aphasia may have slight word-finding problems through to more pervasive problems in understanding spoken language. While some individuals may retain some literacy skills, damage to the language processing centers will usually also affect symbolic representations of language.

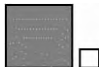
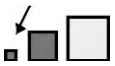
Individuals with congenital language and/or intellectual disabilities (e.g. congenital aphasia and Down's syndrome) may never become literate. The physical inability to speak (e.g. dysarthria resulting from cerebral palsy) may also impact literacy learning, as basic skills required for reading and writing (e.g. phonemic awareness) may be absent.

Visual images, photographs and drawings, provide augmentative and alternative ways to access communication. Such images can be used to enhance text-based interfaces. In addition, sets of symbols (e.g. the Picture Communication System, PCS, see Rebus, see and semantic-based writing systems such as Blissymbolics, see can be used as an alternative to text (Beukelman, & Mirenda, 1998; Wilson, 2003). The type of picture, symbol, or graphic used will depend on the iconicity (ease of recognition), transparency (guessability), opaqueness (logic organization) and learnability of the image. For instance, a photograph of a house may be transparent—for example, it is easily recognizable—while a Blissymbol representing the emotion of happy has logic (heart = feeling; up arrow = up) and is thus opaque and learning (see Table 41.1). The more concrete a representation, the more recognizable it will be. However, representations of abstract meanings, such as emotions, will involve less transparent images necessitating a longer learning curve (see Table 41.1).

Cognitive Prostheses

In addition to providing better access to everyday software, specially designed computer systems have great potential to offer more specific support for people with a wide variety of cognitive impairments. We will describe some specific examples of such systems so that the reader is able to see how such development can proceed. Within the Applied Computing Department at Dundee University, a number of projects have developed computer-based systems to support people with cognitive dysfunction, and examples from the work of this group can il-

TABLE 41.1. Examples of Three Symbol Sets which have been Developed for Augmentative and Alternative Communication—for More Information See About These Symbol Systems See

	PCS	Rebus	Blissymbols
house			
happy			
sad			
big			
small			
fall			

lustrate the range of areas where this technology has particular potential. These examples will also demonstrate the need for appropriate methodologies for research in this area, including need for research into developing more sensitive and effective methodologies.

Software as a Cognitive Scaffolding and a Prompt for Communication

One sequence of projects took as its starting point the improvement of communication systems for physically impaired nonspeaking people. It became apparent that this could be done very effectively by developing models of the cognitive tasks involved in communication. This research has now spawned a new area of development, which is cognitive support for people with dementia, where communicative impairment is just part of their range of difficulties.

For severely physically impaired nonspeaking people, even with current speech output technology, speaking rates of 2 to 10 words per minute are common, whereas unimpaired speech proceeds at 150 to 200 words per minute. In an attempt to improve this disparity, a certain amount of progress has been made in the area of using computers to replace or augment some of the cognitive aspects of communication. Although the cognitive processes underlying language use are incompletely understood, a number of theories that attempt to explain language use have been used to improve the functionality of communication systems for nonspeaking people. This approach to the problem usually involves taking a sociolinguistic view of language. Instead of focusing on the building blocks, or taking a bottom up approach, the interaction as a whole is analyzed, paying attention to its goals, or taking a top down approach to the communication. This may well be a realistic simulation of the natural process, since the production of speech by an unimpaired speaker occurs at such a rate that conscious processing and controlling of the speech at a microlevel is not possible. In common with other learned skills, speech is produced to some extent, automatically, with the speaker being aware of giving high-level instructions to the speech production system, but leaving the details of its implementation to the system.

The nonconscious control of much of speech production has been modeled in the CHAT prototype (Alm, Arnott, & Newell, 1992). This produced quick greeting, farewell, and feedback remarks by giving the user semiautomatic control of exactly what form the remarks would take, within parameters that the user had previously selected. This mimicked the phenomenon of a speaker responding automatically to greetings and other commonly occurring speech routines, without giving the process any detailed thought.

The CHAT-like conversation described illustrates an attempt to achieve a particular communicative goal, achieving social closeness by observing social etiquette. Some recent research efforts have been directed at finding ways to incorporate large chunks of text into an augmented conversation, to help users carry out topic discussion. This has been driven by the observation that a great deal of everyday discourse is reusable in multiple contexts.

Much of this type of discourse takes the form of conversational narratives. Research into the conversational narrative at a sociolinguistic level indicates several interesting characteristics. These include the way in which narratives are told and to whom they are told. For example, a recent event is told repeatedly for a limited time to most people with whom the speaker has contact. As the event recedes in history, the narrative is retold when it is relevant to the topic of conversation. The length of the narrative depends on its age (the older a story is, the more embellished it can become, particularly if it has previously gone down well) and the time available within the conversation. The version of the story (the sequence of events may remain the same while the details or embellishments of a story can differ) depends on factors such as the conversational context and other interlocutors present.

One of the ways to make the retrieval of text chunks easier is to anticipate the chunks that the user may want to use. This has been achieved by modeling the way in which conversational narratives are used using techniques from the fields of artificial intelligence and computational linguistics (Waller, 1992). The prototypes developed constantly adapt to the users' language use, thus mirroring the user's perception of where conversation items are stored. In this way, the system adapts to the way the user thinks instead of having the user learn a new retrieval system.

One of the arguments against using such prediction was raised when word prediction systems were first developed in the early 1980s. Therapists and teachers were concerned that nonspeaking people, especially children, would select what was offered on the screen rather than what they originally wanted to say. Although this may happen, research into predictive systems applied to writing suggests that they may carry over the help they offer and have a wider effect on the users' ability development. Some of this research reports an increase in written output by reluctant writers and people with spelling problems (Newell, Arnott, Booth, & Beattie, 1992). A general improvement in spelling has also been noted. Children with language dysfunction and/or learning disabilities have shown improvement in text composition (Newell, Booth, & Beattie, 1991). This research is in the writing domain, but the results suggest that predictive systems can offer assistance without becoming mere substitutes for creative expression.

Also, it is true of unimpaired conversation that speakers often change direction in their communication depending on chance occurrences, or on the sudden recollection of a point they would like to include. Thus, there is a degree of opportunism in all conversations. Another argument in favor of offering predicted phrases and sequences is that the current situation for most augmented communicators is that their conversations tend to be quite sparse, with control tending to reside with unaided speaking partners. If it is not possible to go boating on the lake, easily going off in any direction you please, is it not preferable to build a boardwalk out over the water than to stay on the shore?

One of the motivations to improve communication systems for nonspeaking people is the fact that they are commonly perceived by people who do not know them as being less intellectually capable than they actually are. It is often reported by nonspeaking people that they are considered unintelligent or immature by strangers. The issue of perceived communicative competence is one that needs increased attention (McKinlay, 1991).

Related to this, an interesting finding emerged from work in which one of the authors was involved. Here, a prototype communication system was used to evaluate listeners' impressions of the content of computer-aided communication based on prestored texts, as compared to naturally occurring dialogues. The nonspeaking user was able to use only prestored texts in order to conduct the conversations. Most of the text was material about one subject (holidays). A number of rapidly accessible comments and quick feedback remarks were also available. The unaided conversations were between pairs of normally speaking volunteers who were asked to converse together on the topic of holidays. Transcripts of randomly sampled sections of the conversations and audio recordings of reenactments of the samples with pauses removed were rated for social competence on a six-item scale (coefficient alpha = 0.83) by 24 judges. The content of the computer-aided conversations was rated significantly higher than that of the unaided samples ($p < .001$). The judges also rated the individual contributions of the computer-aided communicator and the unaided partners on how socially worthwhile and involving these appeared. There was no significant difference between the ratings of their respective contributions ($p > .05$; Todman, Elder, & Alm, 1995).

This finding came as something of a surprise to the researchers, since the original purpose had been to establish whether conversations using prestored material would simply be able to equal naturally occurring conversations in quality of content. Of course, the pauses in actual computer-aided conversation (removed in the above analysis) do have an effect on listeners' impressions of the quality of the communication, but this finding is still of interest, since it suggests that in some ways augmented communication could have an edge over naturally occurring talk. A plausible explanation for this finding is that naturally occurring talk is full of high-speed dysfluencies, mistakes, substitutions, and other messy features that listeners tend to discount with their abilities to infer what the speaker is intending to say. Prestored material is by its nature selected because it may be of particular interest, and it is expressed more carefully than quick flowing talk, and thus may appear more orderly and dense with meaning than natural talk.

In addition to conversational narratives, another common structure in everyday communication is the script, particularly where the speaker is undertaking some sort of transaction. Scripts may be a good basis for organizing prestored utterances to attempt to overcome the problem of memory load when operating a complex communication system based on a large amount of prestored material. Users' memory load can be reduced by making use of their existing long-term memories to help them locate and select appropriate utterances from the communication system. Schank and Abelson (1977) proposed a theory that people remember frequently encountered situations in structures in long-term memory, which they termed *scripts*. A script captures the essence of a stereotypical situation and allows people to make sense of what is happening in a particular situation and to predict what will happen next. Other research (e.g., Vanderheiden, Demasco, McCoy, Pennington, 1992) has shown the potential that similar script-based techniques offer to this field.

An initial experiment was devised (Alm, Arnott, & Newell, 1995) to investigate the potential of a script-based approach to transactional interactions with a communication system, and a prototype system was developed to facilitate this experiment. The aim was to ascertain whether or not a transactional interaction could be conducted using a script-based communication system. It was decided to simulate a particular transactional interaction that could reasonably be expected to follow a predictable sequence of events, for example, one that would be amenable to the script approach, in order to find out whether a computer-based script could enable a successful interaction.

The transaction chosen for the experiment was that of arranging the repair of a household appliance over the telephone. Although the script interface was a relatively simple one devised for the purpose of this experiment, it was successful in facilitating the interaction, and produced a significant saving in the amount of physical effort required.

To take this work further a large-scale project was undertaken to incorporate scripts into a more widely usable device. The user interface of this system is made up of three main components: scripts, rapidly produced speech acts, and a unique text facility. The scripts component is used in the discussion phase of a conversation and consists of a set of scripts with which the user can interact. The rapid speech act component contains high frequency utterances used in the opening and closing portions of a conversation and in giving feedback and consists of groups of speech-act buttons. This facility is based on previous work with CHAT. The unique text component is used when no appropriate prestored utterances are available and consists of a virtual keyboard, a word prediction mechanism, and a notebook facility.

To provide access to a set of scripts, an interface was devised that involved a pictorial representation of the scenes in the script. The pictorial approach was taken in order to give users easier access to the stored material and to assist users with varying levels of literacy skills. In this interface, scripts are presented to the user as a sequence of cartoon-style scenes. The scenes give the user an indication of the subject matter and purpose of the script and assist the user to assess quickly whether the script is appropriate for current needs. Each scene is populated with realistic objects chosen to represent the conversation tasks that can be performed. Thus, the user receives a pictorial overview of the script, what happens in it, and what options are available. This assists the user to see quickly what the script will be able to do in the context of the current conversation. An example of the interface for the system can be seen in Fig. 41.1 below, which shows a scene within the at the doctor script.

Research into picture recognition and memory structures has demonstrated that groups of objects organized into realistic scenes corresponding to stereotypical situations better assist recognition and memory compared to groups of arbitrarily placed objects (Mandler & Parker, 1976; Mandler, 1984). The scene-based interface using a realistic arrangement of objects within a scene was therefore chosen to facilitate recognition and remembering by the user and thus reduce the cognitive load required to locate suitable objects during a conversation.

As it would be impractical to provide scripts for every conceivable situation, it was decided to provide users with a limited number of scripts together with an authoring package with which they can develop their own custom scripts with help from their therapists.

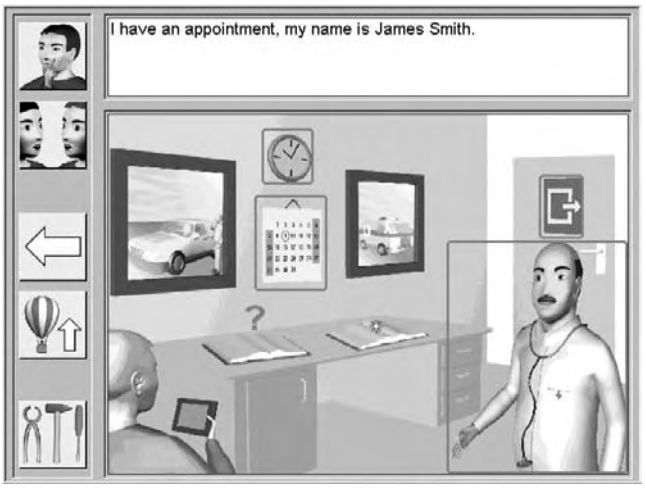


FIGURE 41.1. The script system user interface showing a scene from the doctor script.

A text preview and display box appears at the top of the user interface. The main interface area (bottom right) contains the scene image. The function buttons on the left side of the interface are, from top to bottom: "I'm listening" rapid speech-act button; button to access the main rapid speech-act interface; scene navigation backtrack to previous scenes button; scene navigation overview button; and tool button to access the notepad and additional system control facilities.

It was initially decided to develop six complete scripts. These were chosen after discussions with a user advisory group about situations in which they found difficulty communicating. The scripts developed were at the doctor, at the restaurant, going shopping, activities of daily living, on the telephone, meeting someone new, and talking about emotions.

The system uses the script to guide the user through a dialogue. There is a prediction mechanism, which predicts the next most probable stage in the dialogue that the user will need (based on the script), so the user can usually follow a predicted path through a conversation. This prediction mechanism monitors the sequences of objects selected and uses this information to modify future predictions.

Help for Aphasia

Communication systems for nonspeaking people have been described which in some way model the cognitive processes underlying communication. In the case of most physically disabled nonspeaking people, this is needed in order to speed up the communication process. However, in the case of speech problems that are caused by a stroke or other trauma (aphasia), the person trying to communicate will also have cognitive problems to deal with. Interestingly, the objections that conversation modeling might provide an active prompt to communication suggested a way of possibly helping people who might need such a prompt in order to initiate communication at all.

In a research project investigating the possibility of prompting people with Broca's aphasia in their communications, a predictive communication system was developed with a very simple interface (Waller, Dennis, Brodie, & Cairns, 1998). The system held personal sentences and stories, which were entered with the help of a caregiver. The user could then retrieve the prestored conversational items, with the system offering probable items based on previous use of the system. The interface was designed to be as simple as possible and to be usable by people who were unfamiliar with technology, had language difficulties, but had retained the ability to recognize familiar written topic words.

To access the sentences and stories, the user is led through a sequence of choices on the screen. First, they are offered a choice of conversational partners, a list of topics most likely to be appropriate for the chosen partner, and then a list of the four most common sentences for that partner and that topic. The user can choose to speak one of the sentences through a speech synthesizer, or have the system look for more suggestions. The sentences and topic categories are personal to each user, and the order in which topic words or sentences are presented depends on the past use of the system. Thus, the system is specific to users, both in the information content and in how it adapts to individual ways of communicating.

The system was evaluated with five adults with nonfluent aphasia who were able to recognize but not produce familiar written sentences. There was little change in the underlying comprehension and expressive abilities of the participants while not using the system. When making use of the system, the results showed that some adults with nonfluent aphasia were able to initiate and retain control of the conversation to a greater extent when familiar sentences and narratives were predicted. In other words, users' existing/residual abilities (e.g., small vocabulary, pragmatic knowledge of conversation) were to a degree augmented by the computer functioning as a cognitive prosthesis. This project indicated that a communication system based on prompting could be of help to people with cognitive and communication difficulties.

Support for Nonliterate Users

Children with complex disabilities may have a combination of physical and intellectual impairments that impact on the development of speech and language. Some individuals will learn to read and write using this medium in combination with nonverbal means to communicate. Others will use an alternative graphic medium for interaction. One of the drawbacks of providing symbol-based communication is the difficulty in providing access to novel vocabulary. The ongoing BlissWord project (Andreasen, Waller, & Gregor, 1998) is investigating ways in which users with physical and cognitive limitations can explore new vocabulary. Blissymbolics is a semantic-based natural written (pictographic) language, similar to Chinese. Because of its generative characteristics (BlissWords are spelled using a sequence of one or more Bliss characters), predictive algorithms can be applied to Blissymbolics to assist users in the retrieval of words.

BlissWords are sequenced beginning with a classifier (e.g., all emotions begin with a heart). As illustrated in Fig. 41.2, selecting a shape from the Bliss keyboard, the interface produces a list of BlissWords which begin with classifiers using that shape.



FIGURE 41.2. Screenshot showing Blissymbol keyboard on left and prediction list on right.

Frequency and word lists can be used further to refine the BlissWords that are displayed. Users do not need to be literate to explore language and vocabulary. It is envisaged that video clips and spoken explanation could further augment learning through exploration.

Support for Dementia

Dementia, which involves the loss of short-term memory in elderly people, is a very serious problem for the person and for their families and caregivers. It can rule out most social activities and interactions, since these depend on a working short-term memory for effective participation. This includes even the essential ability to communicate.

As well as being valuable with all older people, reminiscence is an important tool used to help elderly people who have dementia (Sheridan, 1992; Feil, 1993). This is because, while their short-term memories may be impaired, their long-term memories are often more or less intact (Rau, 1993). The difficulty is accessing these long-term memories without the capability of keeping a conversation going, which depends on short-term memory. Activities that do not require patients to maintain the structure of a conversation can help, for instance, looking at and commenting on a series of photographs, which can provide a framework for meaningful person-to-person interaction.

The tools used in such reminiscence work can also include videos, sound, music, and written materials. However, traditionally these are all in separate media, and it can be very time consuming searching for a particular photo, sound, or film clip. Bringing all these media together into a digital multimedia scrapbook could mean easier access to content for more lively reminiscence sessions. The intention of this project was to begin to develop a system that could act as a conversation prosthesis, giving the user the support needed to carry out a satisfying conversation about the past.

An investigation was undertaken to determine which aspects of multimedia would be most helpful for such a reminiscence experience, and the best way to present them. A number of prototype interfaces for a multimedia reminiscence experience were developed. These included text, photographs, videos, and songs from the past life of the city. The materials were collected with the assistance of the University, Dundee City archives, and the Dundee Heritage Project. The prototypes were demonstrated for people with dementia and their care staff at a day center run by Alzheimer Scotland Action on Dementia. The following issues were addressed in these evaluation sessions:

1. *Is it better for the display to use the metaphor of a real-life scrapbook or a standard computer screen?* Six of the staff members preferred the book presentation, three preferred the screen, and two had no preference. Interestingly, this was almost a reversal of the preference shown by the people who had dementia, with the majority preferring the screen presentation. The preference shown for the screen presentation could be due to reduced cognitive ability. The book presentation is a metaphor that may not be interpreted suitably by the person with dementia.
2. *How should the scrapbook material be organized—by subject or by media type?* The majority of the staff evaluators preferred the arrangement by subject saying it was more logical, some were unsure; however, no one showed a preference for the arrangement by media. The clients with dementia reflected these findings. Despite preferring the arrangement being by subject, the majority of evaluators could see benefits from having access to both arrangements. It was concluded that for basic reminiscence sessions the arrangement by subject is preferable. But access to the arrangement by media should be an option, to make the software available for use in other ways.
3. *How do the sounds, pictures, video and, music add to the reminiscence process, and what are their differential effects?* Most of the videos and photographs and all the songs were able to spur conversations. However, it was found that with videos the clients were able only to identify strongly with them when they triggered off specific personal memories, whereas songs and photographs were more generally appreciated. Attention remained focused longest with the songs, which were particularly enjoyed when played repeatedly with everyone singing along. However, the staff felt that some individual clients had enjoyed the videos most.

One general finding was that the multimedia presentation as a whole produced a great deal of interest and motivation from the people with dementia. The staff was also very keen to see the idea developed further. A new project is now beginning which will fully develop the ideas produced by this preliminary work and will produce a fully functional multimedia reminiscence system. The project will be carried out by a multidisciplinary team consisting of a software engineer, a designer, and a psychologist specializing in aging.

In addition to its benefits for the person with dementia, participation in reminiscence activities has been shown to have a positive outcome for the caregivers who take part. Thus, help for reminiscence is not only a tool to stimulate interaction, but also a contributor to improved quality of life, for the person with dementia and his or her family.

Recent work on using videos to present life histories for people with dementia has shown that new technologies, where sen-

sitively and appropriately applied, can bring a substantial added impact to supportive and therapeutic activities for people with cognitive problems (Cohen, 2000). We hope the reminiscence system developed will have the immediate application described above and will also serve as an exploration in developing a range of computer-based entertainment systems for people with dementia.

SYSTEMS TO SUPPORT PEOPLE WITH DYSLEXIA

Dyslexia is a language disorder that is still in the process of being investigated and thoroughly understood. A word-processing environment has been developed to alleviate some of the visual problems encountered by some people with dyslexia when they read and produce text. However, in the absence of a full understanding of the nature of dyslexia, the researchers' approaches were to identify some of the most commonly noted problems which dyslexics. On the basis of these common difficulties, they identified ways in which each individual might be able to minimize the consequences of their own particular problems by manipulating the appearance of their word-processing environment and text presented within it. This stage of the development process involved the start of an iterative cycle of prototype development and evaluation with dyslexic computer users. The work led to a software system which provides a highly (and easily) configurable environment for dyslexic people (who experience a very wide range of differing preferences and problems) to use for reading and producing text. The approach was to examine the parameters of the situation, including cost, cost benefit, existing software, and user demand, with a view to finding an optimal path to the production of a software system which is of real use in practical situations. While users were extensively involved throughout the development process, the researchers did not always rely directly on their individual inputs for further development; rather they ensured that at all times the process was sensitive to user need and opinion, but informed by relevant sources ranging from documented wisdom about the subject to hunches about what might work.

This development of a word processor to assist dyslexics is thus a particularly illuminating example of how the needs of people with a particular type of cognitive impairment can be effectively factored into the design and development process. This research will thus be described in some depth, as an example of a development and to assist the reader in appreciating the generic importance of this approach, which requires knowledge of the underlying syndrome, and a methodology, which encourages an innovative approach to user involvement.

Dyslexia, as has been noted, is still being investigated as a language disorder, and at present, it has a number of definitions. The British Dyslexia Association (2006) offered this description:

> Dyslexia is best described as a combination of abilities and difficulties that affect the learning process in one or more of reading, spelling, and writing. Accompanying weaknesses may be identified in areas of speed of processing, short-term memory, sequencing and organisation, auditory and/or visual perception, spoken language and motor skills. It is particularly related to mastering and using written language, which may include alphabetic, numeric and musical notation.

The symptoms of dyslexia vary greatly and the reasons for the existence of these symptoms are also varied. However, two distinct types of dyslexia are recognized: acquired dyslexia and developmental dyslexia.

Acquired dyslexia is associated with those people who have difficulties caused by damage to the brain. Therefore, prior to the occurrence of brain damage, no difficulties would have been identified. This area can be subdivided into disorders in which the visual analysis system is damaged—peripheral dyslexia—and disorders in which processes beyond the visual analysis system are damaged, resulting in difficulties affecting the comprehension and/or pronunciation of written words—central dyslexia.

Peripheral dyslexia is associated with difficulties such as misreading letters within words and migrating letters between words. Central dyslexia on the other hand concerns issues such as bad comprehension due to an impaired semantic system, and the inability to read unfamiliar words, while familiar words are read easily. Examples would include *monkey* being read as *ape* (a semantic error), and *patient* being read as *parent* (a visual error).

Developmental (or congenital) dyslexia is present from birth but will generally become apparent later, and has been described as follows:

> Developmental dyslexia is a learning disability which initially shows itself by difficulty in learning to read, and later by erratic spelling and by lack of facility in manipulating written as opposed to spoken words. The condition is cognitive in essence, and usually genetically determined. It is not due to intellectual inadequacy or to lack of sociocultural opportunity, or to emotional factors, or to any known structural brain defect. (Critchley, 1964, p. 5)

One of the main features of dyslexia is the individual nature of the disorder. The condition is not typically characterized by one single difficulty, but by a range of difficulties that will vary in combination and in intensity between individuals, giving rise to an enormous variation between individuals in the problems encountered. Each dyslexic person thus has a range of difficulties that need to be addressed differently from those of others. Dyslexia is an example of the need to design for dynamic diversity.

The wide-ranging characteristics of dyslexia provide a challenge for technological assistance, as a single approach will not be appropriate for the range of problems presented by the population of dyslexic people. Computer technology offers the opportunity to provide reading and writing systems that are highly configurable for each individual user but they need to be based on an understanding of the problems which dyslexics have in reading and writing, and some of the visual problems which can affect them. The approach adopted in this research was to offer dyslexic users a range of appropriate visual settings for the display of a word processor, together with the opportunity to configure easily the way in which text is displayed to them. The user can select, by experimentation, the settings that best suit them. These settings are then saved and later recalled each time that person uses the word processor. It will be seen that this approach affords the potential to make computer-based text significantly easier to read than printed text, as well as improving the usability of computer word-processing systems for a wide range of dyslexics.

Some Common Problems of Dyslexia

It was first necessary to determine the parameters which should be offered for configuration by the user. An initial investigation revealed that some of the most commonly encountered problems are as follows (adapted from Willows, Kruk, & Corcos, 1993; D. Shaw, personal communication, May, 1994):

1. *Number and letter recognition.* One of the fundamental problems faced by dyslexics is the recognition of individual alphanumeric symbols. This is often seen when letters that are similar in shape, such as *n* and *h* and *f* and *t,* are confused. The problem is exacerbated with the introduction of uppercase letters. In addition, many dyslexic adults, who are capable of reading printed letters, have difficulty in reading cursive writing.
2. *Letter reversals*. Many dyslexics are prone to reversing letters, which results in a particular letter being interpreted as another letter. Examples of these characters would be *b, d, p,* and *q.* This problem can result in poor word recognition with words containing reversal characters being substituted for other words such as *bad* for *dad*.
3. *Word recognition*. As well as the substitution effect caused because of letter reversals, words that are similar in their outline shape (word contour) can be substituted by dyslexics. Typical examples of this problem are the words *either* and *enter.* Both words have the same start and finishing characters and this, allied with their similar word contours, make them candidates for being substituted for each other when they occur in the text.
4. *Number, letter and word recollection*. Even if the ability to recognize numbers and letters is adequate, it can still prove difficult for a dyslexic individual to recall the actual form and shape of a character. Many dyslexics have so much difficulty recalling upper and lower case characters that they continue to print later in life. Similarly, poor visual memory means that dyslexics have little ability to distinguish whether or not a word looks right.
5. *Spelling problems*. Due to the problems discussed above, dyslexics can have great difficulty with spelling, and many dyslexics have very poor spelling. Much of the spelling of dyslexics appears to reflect a phonic strategy with words like *of* and *all* being spelled *ov* and *ohl.*
6. *Punctuation recognition*. As with characters, dyslexics appear to have difficulty recognizing punctuation marks.
7. *Saccadic and fixation problems*. Another problem that is found in many dyslexics is their lack of ability to follow text without losing their places. Many find it difficult to move from the end of one line to the beginning of the next and find themselves getting lost in the text.
8. *Word additions and omissions*. Dyslexics may add or remove words from a passage of text, apparently at random. This is manifested by words being omitted or duplicated, extra words being added, or word order being reversed or otherwise jumbled.
9. *Poor comprehension*. With the variety of errors caused by the factors described above, a dyslexic person may perceive a totally different (or impoverished) passage of text from the one that is actually in front of them. Dyslexics thus display poor comprehension skills due to text which they perceive being significantly different from the actual text.

Computer Aid for the Problems of Dyslexia

In an attempt to alleviate some of the problems discussed above, dyslexics, particularly within the education system, are encouraged to use computers for text manipulation. The use of a computer keyboard has the potential to alleviate the problems of character recollection, but this only really helps with the recollection of characters, not the recognition of them once they are on the screen.

There is, however, strong evidence to suggest that the use of lexical and spelling aids can greatly assist with spelling problems exhibited by dyslexics (e.g., Newell & Booth, 1991). However, merely highlighting an incorrect word and offering a replacement may not be enough, since one of the other problems that some dyslexics face is an inability to tell if a word looks right, thus they will have difficulty selecting the appropriate corrections.

Some dyslexics are sensitive to color and colored acetate screens or tinted glasses as well as lighting conditions can improve their abilities to read text. Many dyslexics also report interference from peripheral vision, indicating that anything that can be done to reduce screen clutter outside the main screen window, such as making the document page fill the whole screen, may be of benefit (D. Shaw, personal communication, May, 1996).

Based on the above difficulties encountered by dyslexics, the researchers considered ways that the screen image of the text could be manipulated. These included, foreground and background color, character typeface, font and spacing, making letters or words with similar shapes distinguishable by using different fonts, sizes, and/or colors, and presenting text in narrower columns, or with different spacing between words and/or lines to reduce saccade and fixation problems. The researchers investigated potentially promising ideas by implementing prototypes and evaluating their utility with dyslexic users, with an overall view to developing a configuration system that will enable all dyslexics to set up their own optimized environments.

An Experimental Text Reader for Dyslexics

The first stage of the research was to develop an experimental text reader. This prototype presented the user with an easily configurable interface, which allowed for a number of display variables to be altered. Initially, these were background, foreground and text colors, font size and style, and the spacing between paragraphs, lines, words, and characters. The interface was designed in such a way that it gave visual feedback on selections before they were confirmed and made minimal use of text instruction.

This was evaluated using 12 computer literate dyslexic students from higher education using think-aloud techniques, as well as questionnaires and interviews. At various development stages, the helpers were asked to try out the system with a view

to seeing if it was possible for them to put together a display, which improved their abilities to read text from the screen. All the users were able to find a setting that was subjectively superior for them to standard black text on a white background with Times Roman 10- or 12-point text, but the screen layouts that were developed by the test subjects were extremely varied. This highlighted the individual nature of the disorder, and the diverse characteristics of any interface that would be appropriate for this group.

Each appeared to have a favorite color combination, although brown text on a green background was liked by all the testers. Subjects were in greatest agreement about the selection of a typeface: Sans Serif. Arial was rated the best by almost all the testers. All reported that increasing the spacing between the characters, words, and lines was beneficial. The most interesting point that arose during the testing, however, was the fact that at the beginning of the evaluation period the dyslexic subjects did not appear to be aware that altering these variables might be of any use.

A second prototype was then developed based on Word for Windows (Microsoft, 1994; 1995) macros to provide the required configuration interfaces. This was based on the concept of an evolutionary system, rather than a fixed prototype. It was clear that there would be a substantial advantage in developing a dyslexic configuration, but this design decision raised an interesting deviation from the received wisdom of the desirability of WYSIWYG (What You See Is What You Get). In the case of a dyslexic user, what you see should be whatever you can read best, and print previewing facilities would have to be used to show how the layout will appear when printed.

This prototype provided a facility to enhance characters prone to reversals (e.g., *b, d*), by using color font type and size. This idea of coloring reversal characters provided very interesting and unanticipated results, which are described below. Fixation problems were tackled by reducing the page width, and a speech synthesizer that could read the text on the screen was included.

There were two distinct parts to the overall solution, a preference program, and a reading/editing program. The first allowed users to experiment with the various parameters and the second made use of these preferences within a reading and editing environment. The preference program menu presents the user with various options and variables, together with a preview facility, to enable the user to experiment with, and finally store his or her data in a preferences file.

The fact that a unique user environment, tailored to the need of each individual is provided means that the document is (deliberately) not WYSIWYG. A print option thus allows the user to print the document as it appeared with his or her preferred formatting applied to it, or as it would appear without any special formatting.

This second prototype was developed as an add-on module to Microsoft Word, and was evaluated by seven dyslexic users of 15 to 30 years old, in a similar fashion to that above. The users found the system easy and intuitive to use, reporting that each of the options had an effect on their abilities to read. The options that allowed the user to change the color scheme of the document appeared to be the most helpful, but font size and spacing, column width, and indications of reversals were also reported to assist reading by some or all of the users.

The reversals option provided the most interesting results of all. However, the reason for the improvement was not always that the reversal characters were clearly distinguished and easy to read. Instead, it was claimed that the sporadic coloring broke the text up and resulted in the user being less likely to get lost, for example, the system was reducing fixation problems rather than recognition problems.

As the testing progressed, the testers appeared to be surprised at times by the effect some of the changes had on their abilities to read the document. Comments included "I would never have thought of doing that," or "I don't think that will do me much good" before finding that a feature did indeed help.

The prototypes were developed from the perspective that the user population was diverse, and that the design process must accommodate potential changes in preferences over time. The fact that dyslexia is a very idiosyncratic disorder and findings that the users were often unaware of how easy it was to improve their reading potentials by changing visual aspects of the reading environment, illustrates how a standard user-centered design methodology is not appropriate for such user groups.

RESEARCH METHODOLOGIES

The research described above gives a flavor of successful approaches to developing human interfaces and software to support people with various types of cognitive impairment. Much of the methodology used in these developments, however, had to be developed *ab initio*. Traditional user-centered design does not have the flexibility for these user groups, and most research and development in the field of communication and information technology to support people with disabilities has, to date, concentrated on the development of special assistive systems and on accessibility features for younger, mainly physically or sensorially disabled people. Similarly, the human interfaces to most computer systems for general use have been designed, either deliberately or by default, for a typical, younger user (Newell & Cairns, 1993; Newell, 1995; Newell & Gregor, 1997). Knowledge from these fields does not necessarily transfer comfortably to the challenges encompassed in universal design (Beirmann, 1997; Hypponen, 1999; Sleeman, 1998; Stephanidis, 2001) and, in particular, the widely varying and often declining abilities associated with the range of cognitive impairments.

This section addresses the particular issues for the design process which accompany cognitive impairment and suggests a paradigm and methodology to support the process of designing software which is as near to the universal accessibility ideal as is possible, derived from the approach to specific projects described above.

Software systems, which are aimed at the mainstream (rather than being of a prosthetic nature) need to address the wide variation in the types and severity of cognitive impairment between individuals. This demand is further complicated by the fact that as people grow older their abilities change. This process of change includes a decline over time in the cognitive, physical, and sensory functions, and each of these will decline at different rates relative to one another for each individual. This pattern of capabilities varies widely between individuals, and as people

grow older, the variability between people increases. In addition, any given individual's capabilities vary in the short term due, for example, to temporary decrease in, or loss of, function due to a variety of causes, such as the effects of drugs, illness, blood sugar levels, and state of arousal.

This broad range and variability of change presents a fundamental problem for the designers of computing systems, whether they be generic systems for use by all ages, or specific systems to compensate for loss of function. Systems tend to be developed for a typical user and either by design or by default, this user tends to be young, fit, male, and crucially, has abilities that are static over time. These abilities are assumed broadly similar for everybody. Not only is this view wrong, in that it does not take account of the wide diversity of abilities among the wider population of users, but it also ignores the fact that for individuals, these abilities are dynamic over time.

Current software design also typically produces an artifact which is static and which has no, or very limited, means of adapting to the changing needs of users as their abilities change. Even the user-centered paradigm (e.g., ISO 13407, 1999; Nielsen, 1993; Preece, 1994; Shneiderman, 1992) looks typically at issues such as representative user groups, without regard for the fact that the user is not a static entity. Thus, it is important not only to be aware of the diverse characteristics of people with cognitive dysfunction, but also the dynamic aspects of their abilities.

It is clear that people with cognitive impairments, whatever their cause, can have very different characteristics to most human interface and software designers. It is also clear that in these circumstances user-centered design principles need to be employed if appropriate technology is to be developed for this user group (Gregor & Newell, 1999). These methodologies, however, have been developed for user groups with relatively homogonous characteristics. People with dementia, for example, are a diverse group and even small subsets of this group tend to have a greater diversity of functionality than is found in groups of able young people.

An additional complication is that there can be serious ethical issues related to the use of such people as participants in the software development process. Some of these are medically related, but also include, for example, the ability to obtain informed consent. It is thus suggested that the standard methodology of user-centered design is not appropriate for designing for the inclusion of this user group. The importance of research and development taking into account the full diversity of the potential user population, including cognitive diversity was addressed by Newell (1993) in his keynote address to InterCHI '93, where the concept of "Ordinary and Extraordinary Human Computer Interaction" was developed (Newell, 1993; Newell & Cairns, 1993; Newell & Gregor, 1997).

Market share is clearly an important consideration, and this has been given impetus, not only by demographic trends, but also by recent legislation in the United States and other countries, on accessibility of computer systems for people with disabilities. In terms of the workplace, both the Americans with Disabilities Act and the United Kingdom Disability Discrimination Act put significant requirements on employers to ensure that people with disabilities are able to be employed within companies and to provide appropriate technology so that such employees had full access to the equipment and information necessary for their employment. Increasingly, there is political pressure to increase this access, and more requirements for improved access by disabled people are being enshrined in legislation. However, access does not mean only that people with wheelchairs can maneuver around buildings; it also means that there needs to be provision for people with cognitive (and sensory and other physical) impairments to be able to operate computers and other equipment essential to the workplace.

An important additional factor in the value for money equation is that design that takes into account the needs of those with slight or moderate cognitive dysfunction can produce better design for everyone. An example where this has not occurred is illustrated by the problems that the majority of users have had with video tape recorders. If the designers had considered those with cognitive impairments within their user group, they may have been able to design more usable systems. Another example is an e-mail system specifically designed to be simple to use by older people with reduced cognitive functioning, which was found to be preferred by executives to the standard e-mail system that they were used to.

Some people are impaired from birth, but some may become temporarily or permanently disabled by accident or illness (suddenly or more slowly), or even by normal functioning within their employment. This is particularly noticeable in cognitive functioning. Short-term changes in cognitive ability occur with everyone. These can be caused by fatigue, noise levels, blood sugar fluctuations, lapses in concentration, stress, or a combination of such factors, and can produce significant changes over minutes, hours, or days. In addition, alcohol and drugs can also induce serious changes in cognitive functioning, which is recognized in driving legislation, but not in terms of how easy it is to use computer-based systems.

Most people at one time or another, will exhibit cognitive functional characteristics, which are significantly outside the normal range. Although neither they, nor their peers, would consider these people disabled, their abilities to operate standard equipment may well be significantly reduced.

The questions that designers need to consider include:

- Does the equipment that I provide comply with the legislation concerning use by employees who may be cognitively disabled?
- To what extent do I need to take into account the needs of employees who are not considered disabled, but have significant temporary or permanent cognitive dysfunction?
- Should I make specific accommodation for the known reductions in cognitive abilities which occur as employees get older (e.g., less requirement for short-term memory, or the need to learn new operating procedures)?
- What are the specific obligations designers and employers have to provide systems that can be operated by employees whose cognitive ability has been reduced due to the stress, noise, or other characteristics of the workplace?

The argument is that it would be very unusual for anyone to go through their working life without at some stage, or many stages, being significantly cognitively disabled. If equipment designers considered this, it is probable that the effectiveness and efficiency of the work force could be maintained at a higher

level than would be the case if the design of the equipment were based on an idealistic model of the characteristics of the user and their work environment.

The Disabling Environment

In addition to the user having characteristics that can be considered disabled, it is also possible for them to be disabled by the environments within which they have to operate. Newell & Cairns (1993) made the point that the human-machine interaction problems of an able bodied (ordinary) person operating in an high work load, high stress or otherwise extreme (e.g., extraordinary) environment has very close parallels with a disabled (extraordinary) person, operating in an ordinary situation (e.g., an office).

High workloads and the stress levels to which this can lead often reduce the cognitive performance of the human operator. For example, a very noisy environment cannot only create a similar situation to hearing or speech impairment, but can also lead to reduced cognitive performance. The stress level in the dealing room of financial houses can be very high and is often accompanied by high noise levels. A significant advance may be made if the software that was to be used in these houses was to be designed on the assumption that the users would be hearing impaired and have a relatively low cognitive performance. It is interesting to speculate as to whether such systems would produce higher productivity, better decision making and less stress on the operators. Other examples of extreme environments in which people have to operate are the battlefield, under water, or out in space. The stress and fatigue caused by working within such environments means that their performance is similar to that which could be achieved by a very disabled person operating in a more normal environment. It is not always clear that the equipment such people need to operate has been designed with this view of the user.

It is very important to describe the users of technology in terms of their functional abilities related to technology rather than generic definitions of either medical conditions, or primarily medical descriptions of their disabilities. Unfortunately, most statistical data is presented as generic and medically categorizations of disability. Gill and Shipley (1999), however, defined disabled user groups in terms of their functional abilities, with specific emphasis on the use of the telephone. They estimated that within the European Union, which has a population of 385 million, there were 9 million people with cognitive impairment, which could lead to problems using the telephone. Theses figures do not take into account multiple impairments, and the authors pointed out that, in the elderly population in particular, there may be a tendency towards cognitive, hearing, vision, and mobility impairments being present to a varying extent and these may interact when considering the use of technological systems. It is this multiple minor reduction in function (often together with a major disability), which means that the challenges to technological support for older people have significantly different characteristics to that of younger disabled people and to the nondisabled, nonelderly population.

There has been some movement in mainstream research and development in technology, both in academia and industry, away from a technology led focus to a more user led approach, and this has led to the development of user-centered design principles and practices in many industries. In addition, a number of initiatives have been launched to promote a consideration of people with disabilities within the user group in mainstream product development teams with titles including "Universal Design," "Design for All," "Accessible Design," and "Inclusive Design." The "Design for All/Universal Design" movement has been very valuable in raising the profile of disabled users of products, and has laid down some important principles. This approach however has tended not to place too much significance on cognitive impairment, and, particularly if this is included as a factor in the design process, then it becomes more difficult to use traditional user-centered design approaches.

Newell and Gregor (2000) suggested that a new design approach should be developed, which would be based on the already accepted user-centered design methodology. There are some important distinctions between traditional user-centered design with able-bodied users, and the approach needed when the user group either contains, or is exclusively made up of, people with cognitive dysfunction. These include:

- Much greater variety of user characteristics and functionality
- The difficulty in finding and recruiting representative users
- Situations where design for all is certainly not appropriate (e.g., where the task requires a high level of cognitive ability)
- The need to specify exactly the characteristics and functionality of the user group
- Conflicts of interest between user groups, including temporarily able bodied
- Tailored, personalizable, and adaptive interfaces
- Provision for accessibility using additional components (hardware and software)

The balance in the design process also needs to shift from a focus on user needs to one on the users themselves. There will be additional problems when considering people with cognitive dysfunction, which will include

- The lack of a truly representative user group
- That a different attitude of mind of the designer is required
- Ethical issues (Alm, 1994; Balandin & Raghavendra, 1999)
- It may be difficult to get informed consent from some users
- Difficulties of communication with users
- The users may not be able to (sufficiently) articulate their thoughts, or even may be incompetent in a legal sense

Thus, there can be particularly difficult ethical problems when involving users with cognitive impairments in the design process. In addition, it is often necessary to involve clinicians when such users are involved, so some of the user-centered design actually focuses on professional advice about the user, rather than direct involvement of the user. Even with these problems, however, it is possible to include users with cognitive dysfunction sensitively in the design process.

The Inclusion of Users With Disabilities Within Research Groups

In Dundee users with disabilities have a substantial involvement in the research, and they have made a significant contribution both to the research and to the commercial products that have grown from this research. Users are involved in two major ways:

- As disabled consultants on the research team, where they act essentially as test pilots for prototype systems.
- By the traditional user-centered design methodology of having user panels, formal case studies, and individual users who assess and evaluate the prototypes produced as part of the research.

The contribution made by clinicians is also vital to the research, and these are full members of the research team. Dundee's Applied Computing Department is also one of the few computing departments that has employed speech therapists, nurses, special education teachers, linguists, and psychologists (both clinical and cognitive).

User Sensitive Inclusive Design

Some significant changes must be introduced to the user-centered design paradigm if users with disabilities are to be included, and this is particularly important if the users have cognitive impairment. In order to ensure that these differences are fully recognized by the field, the title "User Sensitive Inclusive Design" has been suggested. The use of the term *inclusive* rather than *universal* reflects the view that inclusiveness is a more achievable, and in many situations, appropriate goal than universal design or design for all. *Sensitive* replaces *centered* to underline the extra levels of difficulty involved when the range of functionality and characteristics of the user groups can be so great that it is impossible in any meaningful way to produce a small representative sample of the user group, nor often to design a product that truly is accessible by all potential users.

Design for Dynamic Diversity

In addition to the aspects of user sensitive inclusive design described above, it is necessary to make designers fully aware of the range of diversity which can be expected with cognitively impaired people, and also the changing nature of the cognitive functioning of people. Thus, Gregor and Newell (2000) suggested that this be drawn particularly to the attention of designers by introducing the concept of "Designing for Dynamic Diversity." This process, described above, entails recognition that people's abilities are diverse at any given age and that as they grow older this diversity grows dynamically; it also involves a recognition that any given individual's abilities will vary according to factors such as mood, fatigue, blood sugar levels, and so on. Only by taking on board the factors associated with Designing for Dynamic Diversity will software design produce artifacts which are not static and which have no, or very limited, means of adapting to the changing needs of users as their abilities change.

As has been seen above, metaphors and processes in use at present are limited in meeting the needs of this design paradigm or addressing the dynamic nature of diversity. New processes and practices are needed to address the design issues; awareness raising among the design, economic and political communities has to start; and research is needed to find methods to pin down this moving target.

A story-telling metaphor. In addition, researchers need to consider how best to disseminate the concepts behind universal usability and the results of user sensitive inclusive research. User sensitive inclusive design needs to be an attitude of mind rather than simply the mechanistic application of design for all guidelines. This offers a further challenge to the community. The dangers of using such studies to produce more extensive guidelines has been referred to above, but it is important that the results of user sensitive inclusive design are made available to other designers and researchers. However, it is too early to lay down principles and practices that must be followed by designers, and it may even be impossible to do this for some of the contexts and environments in which designers work. Thus, it is suggested that we follow a story-telling approach, in which information about accessibility issues, and design methods which focus on accessibility is presented in narrative form, with particular examples to illustrate generic principles. This is, in some sense, an extension of the single case-study methodology. This methodology could provide very useful insights to designers in a form that they will find easy to assimilate and act. Thus, this will assist in their educations and will help them to design more accessible products, and better products for everyone.

The use of theater. As an extension of the story-telling metaphor, the research group in Applied Computing at Dundee University has investigated the use of dramatic techniques and theater as a way of addressing the challenges of user sensitive design. For example, as part of the UTOPIA (Usable Technology for Older People: Inclusive and Appropriate) project, they worked in collaboration with the Foxtrot Theater Company to use theater to encourage interaction between (older) users of technology and designers. The outcome was the "UTOPIA Trilogy," a series of short-video plays addressing problems older people have in using technology (Carmichael, A., Newell, Morgan, Dickinson, & Mival, 2005). The films were dramatizations of some of the issues the researchers had encountered during the project. These films were based on real events, conversations, and observations, and they were the amalgamation of many and are intended to convey older people's experiences with technology and the situations they encounter. These videos were evaluated with a variety of audiences including academics, practitioners, software engineers, relevant groups of undergraduates, and older people. This established that the videos provided a useful channel for communication between users of technology and designers, and changed the perceptions of both students and more mature designers of IT systems and products about older people's requirements. A similar technique has also been used in the requirements gathering phase for an IT system designed to monitor older people in case of falls at home (Marquis-Faulkes, McKenna, Gregor, & Newell, 2003).

This research showed that the use of theater can be a very powerful method of encouraging dialogue between various pro-

fessional groups particularly in a clinical environment, for keeping a focus for discussions, and also for providing a channel for communication between users of technology and designers. The researchers view is that the success of this approach was in large part due to the plays being narrative based rather than having a pedagogic style. That is, they illustrated the issues involved within interesting story lines, with all the characteristics of a good narrative—humor, tension, human interest, and antagonists and protagonists. In addition, the quality of the production, having been produced by theater professionals, played a major part in the success of the venture.

Newell, Morgan, Carmichael, and Gregor (2006) discussed the various ways in which actors and theater can play a part in the design process for human-computer interfaces. This could provide a particularly valuable methodology for the design process when the target users have cognitive impairment and thus may not be appropriate for including within standard user-centered design methodologies.

CONCLUSION

Although it is not necessary for human interface designers whose systems may be used by people with cognition impairment to be fully versed in all aspects of cognition, it is important for them to have some background knowledge of the area. They should also be in contact with experts in other disciplines, such as psychology, and have access to appropriate clinical knowledge. In addition, the development of the concept of and a methodology for user sensitive inclusive design, design for dynamic diversity, and development of story-telling methods for communicating results will facilitate researchers in this field and will provide mainstream engineers with an effective and efficient way of including people with disabilities within the potential user groups for their projects. If both of these, can be achieved it will go some way towards providing appropriate technological support for people with cognitive impairment.

References

Alm, N. (1994). Ethical issues in AAC research. In J. Brodin & E. B. Ajessibm (Eds.), *Methodological Issues in Research in Augmentative and Alternative Communication: Proceedings of the Third ISAAC Research Symposium* (pp. 98–104). Sweden: University Press.

Alm, N., Arnott, J. L., & Newell, A. F. (1992). Prediction and conversational momentum in an augmentative communication system. *Communications of the ACM, 35*(5), 46–57.

Alm, N., Morrison, A., & Arnott, J. L. (1995). A communication system based on scripts, plans and goals for enabling non-speaking people to conduct telephone conversations. *Proceedings of the IEEE Conference on Systems, Man & Cybernetics*, Vancouver, Canada, 2408–2412.

Andreasen, P. N., Waller, A., & Gregor, P. (1998). BlissWord—full access to Blissymbols for all users. *Proceedings of the 8th Biennial Conference of ISAAC,* Dublin, Ireland, 167–168.

Bäckman, L., Small, B. J., Wahlin, Å., & Larsson, M. (2000). Cognitive functioning in very old age. In F. I. M. Craik & T. A. Salthouse (Eds.), *The handbook of aging and cognition* (pp. 499–558). City, New Jersey: Lawrence Erlbaum Associates.

Balandin, S., & Raghavendra, P. (1999). Challenging oppression: Augmented communicators' involvement in AAC research. In F. T. Loncke, J. Clibbens, H. H. Arvidson, & L. L. Lloyd (Eds.), *Augmentative and Alternative Communication, New Directions in Research and Practice* (pp. 262–277). London: Whurr.

Beirmann, A. W. (1997). *More than screen deep—Towards an every-citizen interface to the National Information Infrastructure*. Computer Science and Telecommunications Board, National Research Council. Washington DC: National Academy Press.

Beukelman, D. R., & Mirenda, P. (1998). *Augmentative and alternative communication management of severe communication disorders in children and adults* (2nd ed.). Baltimore, MD: Brookes.

British Dyslexia Association (2006, January), http://www.bdaydyslexia.org.uk/facq.html#q1

Carmichael, A. R. (1999). *Style Guide for the Design of Interactive Television Services for Elderly Viewers*. Winchester: Independent Television Commission.

Carmichael, A., Newell, A., Morgan, M., Dickinson, A., & Mival, O. (2005). Using theatre and film to represent user requirements. *Proceedings INCLUDE '05*, Royal College of Art, London. ISBN (CD rom) 1-905000-10-3.

Charness, N., & Bosman, E. A. (1994, January–March). Age-related changes in perceptual and psychomotor performance: Implications for engineering design. *Experimental Aging Research, 20*(1), 45–61.

Cohen, G. (2000). Two new intergenerational interventions for Alzheimer's disease patients and their families. *American Journal of Alzheimer's Disease, 15*(3), 137–142.

Critchley, M. (1964). *Developmental dyslexia*. London: Heinemann.

Detterman, D. K., Gabriel, L. T., & Ruthsatz, J. M. (2000). Intelligence and mental retardation. In R. J. Sternberg (Ed.), *Handbook of Intelligence* (pp. 141–158). Cambridge: Cambridge University Press.

Feil, N. (1993). *The Validation Breakthrough*. Maryland: Health Professions Press.

Freudenthal, A. (1999). *The design of home appliances for young and old consumers*. Delft: Delft University Press.

Gill, J., & Shipley, T. (1999). *Telephones, what features do people need*. London: Royal National Institute for the Deaf.

Gould, S. J. (1997). *The mismeasure of man* (2nd ed.). Harmondsworth: Penguin.

Gregor, P., & Newell, A. F. (1999). The application of computing technology to interpersonal communication at the University of Dundee's Department of Applied Computing. *Technology and Disability, 10*, 107–113.

Horn, J. L., & Cattell, R. B. (1967). Age differences in fluid and crystallised intelligence. *Acta Psychologica, 26*, 107–129.

Hypponen, H. (1999). *The Handbook on Inclusive Design for Telematics Applications*. Helsinki, Finland: European Union INCLUDE Project.

ISO 13407. (1999). *Human-centred design processes for interactive systems*. International Organisation for Standards.

Kaluger, G., & Kolson, C. L. (1987). *Reading and Learning Disabilities* (2nd ed.). Columbus, Ohio: Bell & Howell Company.

Kaufman, A. S. (2000). Tests of intelligence. In R. J. Sternberg (Ed.), *Handbook of Intelligence* (pp. 445–476). Cambridge: Cambridge University Press.

Lohman, D. F. (2000). Complex information processing and intelligence. In R. J. Sternberg (Ed.), *Handbook of Intelligence* (pp. 285–340). Cambridge: Cambridge University Press.

Marquis-Faulkes, F., McKenna, S. J., Gregor, P., & Newell A. F. (2003). Scenario-based drama as a tool for investigating user requirements with application to home monitoring for elderly-people. In D. Harris, V. Duffy, M. Smith, & C. Stephanidis (Eds.), *Human-Centred Computing: Vol. 3. Cognitive, Social and Ergonomic Aspects* (pp. 512–516). Mahwah, NJ:Lawrence Erlbaum.

McKinlay, A. (1991). Using a social approach in the development of a communication aid to achieve perceived communicative competence. In J. Presperin (Ed.), *Proceedings of the 14th Annual Conference of the Rehabilitation Engineers Society of North America* (pp. 204–206). Washington, DC: The RESNA Press.

Mandler, J. M. (1984). *Stories, scripts and scenes: Aspects of schema theory*. Mahwah, NJ: Lawrence Erlbaum Associates.

Mandler, J. M., & Parker, R. E. (1976). Memory for descriptive and spatial information in complex pictures. *Journal of Experimental Psychology: Human Learning and Memory, 2*, 38–48.

Microsoft, (1994). *Word Developers Kit.* Redmond, Washington: Microsoft Press.

Microsoft. (1995). *Word for Windows 95*. Redmond, Washington, Microsoft Corporation.

Newell, A. F. (1995). Extra-ordinary human computer operation. In A. D. N. Edwards (Ed.), *Extra-ordinary human-computer interaction* (pp. 3–18). Cambridge: Cambridge University Press.

Newell, A. F., Arnott, J. L., Booth, L., & Beattie, W. (1992). Effect of the PAL word prediction system on the quality and quantity of text generation. *Augmentative and Alternative Communication, 8*, 304–311.

Newell, A. F., & Booth, L. (1991). The use of lexical and spelling aids with dyslexics. In C. Singleton (Ed.), *Computers & literacy skills* (pp. 35–44). Hull: University of Hull.

Newell, A. F., Booth, L., & Beattie, W. (1991). Predictive text entry with PAL and children with learning difficulties. *British Journal of Educational Technology, 22*, 23–40.

Newell, A. F., & Cairns, A. Y. (1993, October). Designing for extra-ordinary users. *Ergonomics in Design* (*Human Factors and Ergonomics Society*), 10–16.

Newell, A. F., & Gregor, P. (1997). Human computer interfaces for people with disabilities. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 813–824). Amsterdam: Elsevier.

Newell, A. F., & Gregor, P. (2000). User sensitive inclusive design—in search of a new paradigm. *Proceedings of the ACM Conference on Universal Usability*, Washington, DC, 39–44.

Newell, A. F., Morgan, M. E., Gregor, P., & Carmichael, A. *CHI 2006* (2006). Experience Report in CHI 2006 extended abstracts on Human Factors in Computing Systems Montreal, Quebec, Canada, 22–27 April 2006. pp. 111–117.

Nielsen, J. (1993). *Usability engineering*. London: Academic Press.

Ogozalec, V. Z. (1997, March). A comparison of the use of text and multimedia interfaces to provide information to the elderly. *Proceedings of CHI '97,* Atlanta, Georgia, New York: ACM Press, 65–71.

Pirkl, J. J. (1994). *Transgenerational design, products for an aging population*. New York: Van Nostrand Reinhold.

Preece, J. (1994). *A Guide to Usability—Human Factors in Computing*. London: Addison Wesley & Open University.

Rabbitt, P. M. A. (1993). Does it all go together when it goes? The nineteenth Bartlett Memorial lecture. *The Quarterly Journal of Experimental Psychology, 46A*(3), 385–434.

Rau, M. T. (1993). *Coping with Communication Challenges in Alzheimer's Disease*. California: Singular Publishing Group Inc.

Salthouse, T. (1985). *A Theory of Cognitive Aging*. Amsterdam: North Holland.

Salthouse, T. A. (1991). *Theoretical Perspectives on Cognitive Aging*. Mahwah, NJ: Lawrence Erlbaum Associates.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sheridan, C. (1992). *Failure-free Activities for the Alzheimer's patient*. London: Macmillan Press.

Shneiderman, B. (1992). *Designing the user interface: strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.

Sleeman, K. D. (1998). Disability's new paradigm, implications for assistive technology and universal design. In I. Placencia Porrero & E. Ballabio (Eds.), *Improving the Quality of Life for the European Citizen: Vol. 4 Assistive Technology Research Series* (pp. xx–xxiv). Amsterdam: IOS Press.

Stephanidis, C. (Ed.). (2001). *User interfaces for all*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sternberg, R. J. (2000). The concept of intelligence. In R. J. Sternberg (Ed.), *Handbook of Intelligence* (pp. 3–15). Cambridge: Cambridge University Press.

Todman, J., Elder, L., & Alm, N. (1995). Evaluation of the content of computer-aided conversations. *Augmentative and Alternative Communication, 11*(4), 229–234.

Vanderheiden, P. B., Demasco, P. W., McCoy, K. F., & Pennington, C. A. (1996). A preliminary study into schema-based access and organization of re-usable text in AAC. *Proceedings of the RESNA '96 Conference, Salt Lake City, Utah,* USA, 59–61. Arlington, VA: RESMA Press.

Waller, A. (1992). *Providing narratives in an augmentative communication system.* Unpublished doctoral dissertation, University of Dundee, Dundee, Scotland, UK.

Waller, A., Dennis, F., Brodie, J., & Cairns, A. (1998). Evaluating the use of TalksBac, a predictive communication device for nonfluent adults with aphasia. *International Journal of Language and Communication Disorders, 33*(1), 45–70.

Willows, D. M., Kruk, R. S., & Corcos, E. (1993). *Visual processes in reading and reading disabilities.* Mahwah, NJ: Lawrence Erlbaum Associates.

Wilson, A. (Ed.). *Communicating with Pictures and Symbols.* CALL Centre publications (Augmentative and Alternative Communication in Practice: Scotland), University of Edinburgh. Edinburgh. ISBN 1-898042-25-X

Worden, A., Walker, N., Bharat, K., & Hudson, S. (1997). Making computers easier for older adults to use. In *Proceedings of CHI '97* (pp. 266–828).

·42·

PHYSICAL DISABILITIES AND COMPUTING TECHNOLOGIES: AN ANALYSIS OF IMPAIRMENTS

Andrew Sears
UMBC

Mark Young
Maryland Rehabilitation Center

Jinjuan Feng
Towson University

Introduction **830**
Defining Impairment **830**
Physical Impairments and Computing **831**
Health Condition-Induced Impairments **832**
ALS (Amyotrophic Lateral Sclerosis) 832
Arthritis 832
Brain Injury 832
Cerebral Palsy 833
Missing Limbs or Digits (Amelia or Amputation) 833
Multiple Sclerosis 833
Muscular Dystrophy 833
Parkinson's Disease 833
Repetitive Stress Injury 833
Stroke 833
Spinal Cord Injury 834
Tremors 834
Locked-In Syndrome 835
Summary 835
Situationally-Induced Impairments and Disabilities **835**
HCI Research and PI **835**
Spinal Cord Injuries (SCI) 837
Cerebral Palsy (CP) 838
Impairments of the Hands and Arms 839
Impairments in Infants 840
Significant Speech and Physical Impairments (SSPI) . . 840
Input Using Electrophysiological Data 841
Cursor-Control Technologies 842
Head 842
Tongue 843
Eye and eyelid 843
Speech 844
Text Entry Technologies 844
Keyboard-based text entry 844
Speech-recognition-based text entry 845
Eye-gaze-based typing 846
Adoption or Abandonment of Assistive Technologies **846**
Conclusions **847**
PI Does Not Imply Disability 847
PI Affect Cognitive, Perceptual, and Motor Activities . . 847
Basic Actions Can Be Difficult 848
Standardized Descriptions of Health Conditions and PI 848

Successful Adoption of Assistive Technologies Is Critical 848
Future Directions **848**
Acknowledgments **849**
References **849**

INTRODUCTION

Computing devices are becoming smaller, more mobile, more powerful, more ubiquitous, and less expensive, changing the way people work, shop, communicate, and entertain. As a result, an inability to use these technologies can significantly limit employment, educational, and social opportunities. While computing devices can be convenient tools for traditional computer users, they can also create barriers for individuals with physical, cognitive, or perceptual impairments. For example, individuals with certain types of tremors may experience difficulty using small keyboards. However, a design process that considers the impairments of users can turn these potential barriers into powerful tools that increase employment opportunities, provide enhanced communication capabilities, and enable increased independence (Young, Tumanon, & Sokal, 2000).

While cognitive, perceptual, and physical impairments (PI) can all hinder the use of computing technologies, this chapter focuses on specific physical impairments that contribute to disability. More specifically, we focus on PI that may hinder an individual's ability to physically interact with computing technologies. Therefore, we do not address PI that affect the lower body or PI that hinder the production of speech. For additional information regarding cognitive or perceptual impairments, see chapters 41, 43, and 45. Chapters 39 and 40 may also provide useful insights as they discuss the design of technologies for older adults and children, two groups whose physical capabilities may require special attention.

The objectives of this chapter are to:

- Provide an introduction to specific PI that can hinder the use of traditional computing devices;
- Highlight critical characteristics of PI that must be considered when designing computing systems;
- Discuss the relationship between PI that result from health conditions and those that result from environmental or contextual factors;
- Summarize existing HCI research involving individuals with PI;
- Offer observations based upon the literature to guide future research and development efforts.

We believe that understanding PIs will allow for the design of more effective computing technologies that lessen or even eliminate the associated disabilities. We begin by presenting a set of definitions for *impairment*, *disability*, and *handicap* that are offered by the World Health Organization (WHO). By formally defining these terms, and using the terms in a way that is consistent with these definitions, we hope to eliminate potential ambiguity and confusion. Subsequently, we define the subset of PI that is of primary concern when interacting with computing technologies. Through this definition, we refine the scope of this chapter.

Given these definitions, we proceed to discuss the relationship between health conditions and PI. Understanding the underlying health condition is critical since this often provides valuable insights regarding the nature of the resulting impairments. We describe common health conditions (e.g., cerebral palsy, spinal cord injuries) associated with PI that affect interactions with computing devices. We identify the associated PI, important characteristics of these PI, and any additional impairments that may prove critical when designing computing technologies. While health conditions are most often associated with PI, both contextual and environmental factors can also hinder interactions with computing devices. Building on this observation, we briefly discuss the relationship among the environment, context, physical impairments, and disabilities. Most importantly, we highlight similarities and differences between those PI associated with health conditions and difficulties individuals may experience as a result of the environment or the activities in which they are engaged.

We conclude by reviewing recent research, discussing existing technologies, highlighting some challenges associated with adopting these technologies, and offering directions for additional research.

DEFINING IMPAIRMENT

The World Health Organization published the "International Classification of Impairments, Disabilities, and Handicaps" (ICIDH) in 1980. The current version of this classification serves as a foundation for this discussion (WHO, 2006). The ICIDH model acknowledges the complex relationships that exist among health conditions, impairments, disabilities, and handicaps while highlighting the potentially important role of both the context and environment in which activities are taking place. Figure 42.1 highlights the relationships between user capabilities and the influence of both health conditions and context. Both health conditions and context can directly affect the users' capabilities. Often, this influence is gradual rather than abrupt, causing an individual's capabilities to move along

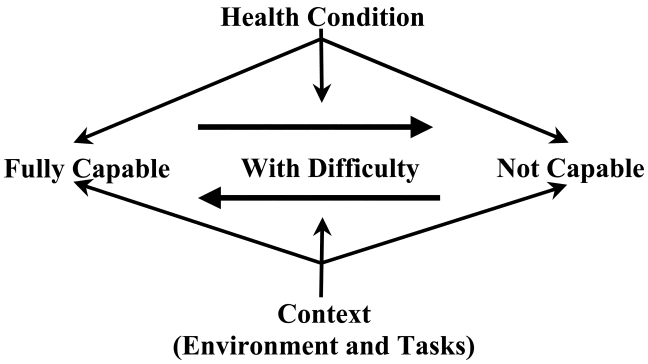


FIGURE 42.1. Model illustrating the relationship between health conditions, context, and an individual's capabilities.

a continuum from fully capable to the point at which they are no longer able to effectively perform a particular task. A specific health condition may, or may not, negatively affect an individual's capabilities, depending on the environment in which the individual is located and the tasks in which they are engaged.

The following definitions are adapted from the ICIDH:

- Health condition: A disease, disorder, injury, or trauma. Examples include spinal cord injuries, arthritis, cerebral palsy, stroke, multiple sclerosis, and amyotropic lateral sclerosis.
- Impairment: A loss or abnormality of body structure or function. Physical impairments include a loss of muscle power, reduced mobility of a joint, uncontrolled muscle activity, and absence of a limb. Impairments can be caused by health conditions. For example, an individual with arthritis (health condition) may experience weakness, stiffness, and a reduced range of motion (impairments). Impairments can also be caused by context. For example, a cold environment can limit joint mobility.
- Disability (activity limitations): Difficulties an individual may have in executing a task or action. Examples include difficulty communicating, learning, performing tasks, or using computing technologies. Disabilities can be caused by impairments; an individual has a disability; disabilities are activity-specific. For example, a loss of muscle power in both arms (impairment) may interfere with an individual's ability to use a keyboard and mouse (disability). Context can also lead to disabilities. For example, working in a moving vehicle does not necessarily result in an impairment (e.g., the individual's capabilities have not been altered), but can result in a disability (e.g., the individual may not be able to use a mouse due to the movement of the vehicle).
- Handicap (participation restrictions): Problems an individual may experience while involved in life situations. Handicaps can be caused by disabilities (which originate with either a health condition or context); an individual experiences a handicap; handicaps occur at a social level. For example, an individual may have difficulty communicating (disability) which places restrictions on her ability to participate in educational and work activities, participate in the exchange of information, and maintain social relationships (handicaps).

Given these definitions the distinction between a disability and a handicap is, in many ways, dependent upon the individual. If an individual has an impairment (e.g., difficulty bending his fingers) that makes it difficult to use a standard keyboard, then this individual has a disability with respect to the operation of the keyboard. However, this individual only experiences a handicap if the inability to use the keyboard interferes with his normal life activities. Our focus is on designing computing technologies to lessen or eliminate disabilities that may result from either PI or context. As a result, the potentially subtle differences between disabilities and handicaps are not important.

PHYSICAL IMPAIRMENTS AND COMPUTING

Impairments, as described by the WHO, can affect every structure and function of the body. In this chapter, we focus on the subset of PI that are most relevant in the context of physical interactions with computing technologies. Impairments, even PI, that interfere with vision, hearing, and other activities involved in the use of computing devices are not discussed (see chapters 41, 43, and 45 for coverage of some of these issues). Four categories of PI are particularly relevant:

- Structural deviations—Situations in which there is significant deviation or loss with regard to anatomical parts of the body. Examples include the partial or total absence of a body part (e.g., missing finger, hand, arm) as well as situations in which a body part deviates from the norm in either position or dimension.
- Mobility (of bone and joint) functions—Mobility functions address an individual's ability to move a joint or bone. This includes both the range and ease of movement. For example, an individual may have limited range of motion or experience difficulty bending her fingers (e.g., arthritis).
- Muscle power functions—Muscle power functions address an individual's ability to generate force by contracting a muscle or muscle group. Paresis (e.g., weakness) refers to the partial loss of power while paralysis refers to the complete loss of power. Both can be caused by various conditions (e.g., brain injury, spinal cord injuries). Muscle tone and endurance functions may also be important, but tend to be less critical for most interactions with computing technologies.
- Movement functions—While additional movement functions exist, we focus on an individual's ability to control voluntary and involuntary movements. Difficulty controlling movements that involve a rapid change in direction (e.g., dysdiadochokinesia) is an example of a problem with a voluntary movement. Uncontrolled shaking or trembling of the hands (e.g., essential tremors or tremors associated with Parkinson's Disease) are examples of involuntary movements.

While there are many other types of PI, including the stability of the joints, motor reflex functions, and gait pattern functions, the four categories identified earlier account for the majority of PI that hinder the use of computing technologies. As we discuss various health conditions, it is useful to identify specific PI that may occur as well as important characteristics of these PI that may influence the design of alternative strategies or technologies. While PI vary in many ways, the following four dimensions are perhaps the most critical:

- PI can be *permanent* or *temporary*;
- PI may be *continuous* or *intermittent*. For example, individuals with Multiple Sclerosis experience relapses and remissions;
- PI may be *progressive* (getting more severe with time), *regressive* (getting less severe with time), or *static* (no change with time);
- The severity of a PI can range from *mild* to *severe*. In other cases, severity is *variable*.

Temporary PI can be challenging because individuals are unlikely to invest significant time or effort in adapting to new interaction techniques. If a PI is *permanent*, individuals will develop accommodation or adaptive strategies and learning new

interaction techniques may be acceptable. *Intermittent* PI create difficulties because alternative interaction techniques must be accessible when needed without hindering interactions when the PI is not present. *Progressive* and *regressive* PI are most effectively addressed by accommodations that adapt as the PI changes. This adaptation may be automatic or user directed. Adaptation is also useful when the severity of the PI is *variable*; the solution can be more complex since severity can change at any time. PI that are permanent, continuous, static, and of a fixed severity are perhaps the easiest to address. Under these conditions, the individuals are likely to adapt to their impairment, develop their own accommodation strategies, and be more willing to invest time learning new interaction techniques.

HEALTH CONDITION-INDUCED IMPAIRMENTS

It is tempting to view all individuals with a particular PI as benefiting equally from a given accommodation. Unfortunately, this perspective can result in solutions that fail to adequately address the needs of the individual. Understanding the underlying health condition provides valuable insights into the nature of the associated PI, which in turn provides insights into the characteristics of the accommodations that will prove most beneficial.

We briefly describe some of the more common diseases, disorders, and injuries associated with PI that hinder the use of computing systems. For additional information regarding any of the underlying health conditions, we refer the reader to the Merck Manual (Berkow, 1997) or the references included throughout this chapter. Our descriptions assume that appropriate treatments (e.g., medication, surgery, therapy) have already taken place. While we do not address issues involved in perceptual, speech, or cognitive impairments, we do highlight situations in which these impairments may be present. We do this for two reasons: (a) to provide additional insights into difficulties individuals with specific health conditions may face (e.g., vision, hearing, or cognition difficulties), and (b) to highlight situations in which specific alternative interaction strategies or technologies may not be appropriate (e.g., speech recognition).

ALS (Amyotrophic Lateral Sclerosis)

ALS is also known as Lou Gehrig's Disease (Taylor & Lieberman, 1988). ALS is a progressive disease that begins with weakness in the hands (less often the weakness begins in the feet). ALS progresses at a variable rate, but 50% of those affected die within 3 years of the initial symptoms and only 10% live beyond 10 years. As the disease progresses, the weakness spreads to additional muscles and becomes more severe (eventually leading to paralysis throughout the body). With time, spasticity occurs resulting in muscles becoming tight. Spasms and tremors can also occur. ALS only affects those nerves that stimulate muscle action. Cognitive functions and sensation remain intact. ALS is characterized primarily by muscle power PI (weakness progressing to paralysis), but movement PI also occur (stiff muscles, spasms, and tremors). The PI are permanent, continuous, and progressive. Initially, the PI are mild, but with time they become severe. Eventually, the muscles involved in producing speech are affected.

Arthritis

Arthritis refers to the inflammation of a joint (Hicks, 1988; Schumacher, 1993). The two most common forms of arthritis are osteoarthritis and rheumatoid arthritis. *Osteoarthritis* (OA) is a chronic joint disorder that can cause joint pain and stiffness. OA is characterized by degeneration of joint cartilage and adjacent bone, is the most common joint disorder, and is one of the most common causes of physical impairments (Berkow, 1997). OA is not an inevitable part of aging, but does occur more often in older populations. OA typically progresses slowly after symptoms first appear. OA can result in significant mobility PI (reduced range of motion, weakness, and difficulty with repetitive motions), which tend to be permanent, intermittent or continuous, progressive, and can range from mild to severe.

Rheumatoid arthritis (RA) is an autoimmune disease that can cause swelling, pain, and often the eventual destruction of the joint's interior. RA is characterized by symmetric inflammation of the joints. RA typically appears first in people between 25 and 50 years of age, affecting 1% of the population. Most often, RA starts subtly, progressing at a highly variable rate. In some rare cases, RA spontaneously disappears and treatment is often successful, but RA can result in significant mobility PI (reduced range of motion, weakness, and difficulty with repetitive motions). The PI tend to be permanent, continuous, and progressive. The severity of the PI is typically mild initially, but in 10% of the cases will progress to severe.

Brain Injury

Brain Injury (BI) is a term used to describe a collection of injuries (Horn & Zasler, 1996; Rosenthal, Griffith, Bond, & Miller, 1990). Technically, there is a disconnection between the cause of the injury (e.g., penetration of the skull by a foreign object, or the rapid acceleration or deceleration resulting from a forceful blow to the head or vehicular accident) and the injury itself. Common head injuries include skull fractures, cerebral contusions (bruises on the brain), cerebral lacerations (torn brain tissue), concussions (brief loss of consciousness after an injury), and intracranial hematomas (collection of blood in the brain or between the brain and skull). Head injuries are the most common cause of death and disabilities for individuals under 50 years of age. BI are not always the result of trauma. For example, anoxia can also result in BI. The consequences of a brain injury depend upon the area of the brain affected and the severity of the injury, but may include death; language, vision, and motor-control difficulties; and periodic headaches. Other parts of the brain often assume the responsibilities of the damaged portion of the brain, lessening the severity of the resulting impairments. This ability is more prevalent in children, where the brain is more adept at shifting functions to different parts of the brain. The most common PI are movement oriented (e.g., difficulty controlling muscles). After rehabilitation, the PI are considered permanent, continuous, and static. Severity may range from mild to severe.

Cerebral Palsy

Cerebral Palsy (CP) is a condition that results from brain injury that typically occurs before, during, or shortly after birth. CP is not a disease and is not progressive (Molnar, 1992). Over 90% of those affected with CP survive to adulthood. There are four main types of CP: spastic, choreoathetoid, ataxic, and mixed. *Spastic CP* occurs in approximately 70% of those individuals with CP and is characterized by both movement and muscle power PI (stiff muscles and weakness). *Choreoathetoid CP* is characterized by movement PI (spontaneous, slow, uncontrolled muscle movements; abrupt and jerky movements) and occurs in 20% of those individuals with CP. *Ataxic CP* occurs in approximately 10% of those individuals with CP and is characterized by movement and muscle power PI (poor coordination, weakness, trembling; rapid or fine movements can be difficult). *Mixed CP* occurs in many individuals and is characterized by a combination of two of the previously mentioned forms. Seizures are also possible (most often with Ataxic CP). In all forms of CP, the PI are permanent, continuous, static, and may range from mild to severe. It is important to note that speech and intelligence can be affected in all forms of CP.

Missing Limbs or Digits (Amelia or Amputation)

Technically, missing limbs or digits is an impairment, not a health condition (Banerjee, 1982). However, like tremors, it is possible for limbs or digits to be missing without any associated health condition (e.g., congenital absence of a limb or digit is not considered a health condition). The PI associated with missing limbs or digits is permanent, continuous, static, and can vary from mild to severe depending on the use and effectiveness of prosthetic devices.

Multiple Sclerosis

Multiple Sclerosis (MS) is a disorder in which the nerve fibers associated with the eye, brain, and spinal cord lose patches of myelin (a protective, insulating sheath) (Ropper & Brown, 2005). MS is progressive, but no single pattern typifies the progression. MS results in numerous symptoms, including a variety of movement and muscle power PI (clumsiness, tremors, stiff muscles, weakness). MS often results in periods of flare-ups alternating with periods of relatively good health (e.g., relapses and remissions), but in some individuals the symptoms become more severe within weeks or months. As flare-ups become more frequent, the symptoms become more severe and may become continuous. The PI are permanent, intermittent, and progressive. The PI may be mild initially, but often progress to become severe. In addition to multiple PI, MS can result in visual impairments.

Muscular Dystrophy

Muscular Dystrophy (MD) is a group of inherited muscle disorders (Pidcock & Christensen, 1997). The two most common forms of MD are Duchenne's MD (affecting 20–30 of every 100,000 boys) and Becker's MD (affecting three of every 100,000 boys). Duchenne's MD is more severe, appearing between the ages of three and seven; most children must use a wheelchair by 10 or 12 and death is common by age 20. Becker's MD is less severe, with symptoms appearing around age 10; very few children are confined to a wheelchair by age 16, and more than 90% live beyond age 20. Both forms begin with muscle power PI (weakness) that spreads and becomes more severe. This is often followed by mobility PI (joints that cannot be fully extended due to contracted muscles). Some PI may not be present early (e.g., inability to fully extend the elbows). In both forms, the PI are permanent, continuous, and progressive. Typically, the PI are mild initially and progressively get worse.

Parkinson's Disease

Parkinson's Disease (PD) is a progressive degenerative disorder of the nervous system (Duvaisin, 1991). PD is characterized by degeneration of the nerve cells in the basal ganglia which results in reduced production of dopamine (the main neurotransmitter of the basal ganglia). PD affects approximately one in 250 individuals over the age of 40, and one in 100 individuals over the age of 65. PD typically begins with mild symptoms and progresses gradually. PD often begins with resting tremors (movement PI). As the disease progresses, initiating a movement can become difficult, muscles become stiff, and bending the elbow can become difficult or uncomfortable (movement and mobility PI). The tremors vary dramatically. Some individuals never develop tremors. For others, tremors may be permanent or temporary; continuous or intermittent; progressive, regressive, or static; and can vary from mild to severe. Other PI, including difficulty initiating movements and stiffness of muscles, are permanent, continuous, and progressive. They are mild initially, but progress to become severe. Still other PI, such as difficulty bending the elbow, may never occur. Individuals with PD often speak softly in a monotone and may stutter.

Repetitive Stress Injury

Repetitive Stress Injuries (RSI) are known by a variety of names including "cumulative trauma disorders" and "repetitive trauma syndromes" (Moore & Garg, 1992). RSI refers to a collection of disorders or injuries that are believed to be associated with repetitive activities, but the precise cause is still a subject of debate. RSI are most commonly associated with the wrists, but can affect other parts of the body. RSI most often result in mobility PI (reduced range of motion, weakness, and difficulty with repetitive motions) that are permanent, continuous or intermittent, progressive, and can vary from mild to severe. A classic example is carpal tunnel syndrome.

Stroke

A stroke is defined as the death of brain tissue resulting from a lack of blood flow and insufficient oxygen to the brain (Wade, Langton Hewer, Skilbeck, & David, 1985). A stroke typically

occurs when a blood vessel becomes blocked or bursts. The symptoms depend upon the area of the brain that is affected, but can include vision, speech, cognitive, and physical impairments. Since other parts of the brain can assume the responsibilities of the damaged portion of the brain, it is possible to lessen the severity of the impairments that are apparent immediately following a stroke. Rehabilitation services are critical in this respect. The most common PI are movement oriented (e.g., difficulty controlling muscles). After rehabilitation, the PI can be defined as permanent, continuous, and static. The severity of the PI may be range from mild to severe.

Spinal Cord Injury

Spinal Cord Injuries (SCI) occur when the spinal cord (a collection of nerves extending from the base of the brain through the spinal column) is compressed, cut, damaged, or affected by disease (Stiens, Goldstein, Hammond, & Little, 1997). The spinal cord contains motor nerves (controlling movement) and sensory nerves (providing information about temperature, pain, position, and touch). The consequences of an SCI depend upon the level and completeness of the injury.

The level of an injury is based upon the nerves that are affected. The spinal column is divided into four areas: cervical (neck), thoracic (chest), lumbar (lower back), and sacral (tail bone). There are seven cervical vertebrae, numbered C1 through C7 from top to bottom. There are 12 vertebrae in the thoracic region (T1–T12), five in the lumbar region (L1–L5), and five in the sacral region. The level of an SCI refers to the location of the damaged nerves. In the cervical region, injuries are labeled based upon the vertebrae immediately below the damaged nerves (e.g., level C1 refers to damage to the nerves just above vertebrae C1). Damage to nerves just below C7 and above T1 are referred to as "level C8." In all other regions, injuries are labeled based upon the vertebrae immediately above the damaged nerve (level T3 refers to injuries to the nerves between T3 and T4). In general, injuries higher on the spinal cord will result in greater impairment. Table 42.1 summarizes the possible consequences of high-level SCI.

The completeness of an injury is often assessed using a scale defined by the American Spinal Injury Association (ASIA, 2001). While there are additional details, and assessments must be completed by trained professionals, the following summaries provide sufficient detail for our purposes. Table 42.2 provides a brief description of the ASIA scores.

SCI result in muscle power PI with the level of the injury determining which muscles are affected and the completeness of the injury determining how much muscle power is lost. Injuries with ASIA scores of A or B result in a complete loss of muscle power (e.g., paralysis) in the affected muscles. Injuries assigned ASIA scores of C or D are more difficult to describe since the amount of residual muscle power can be highly variable (e.g., weakness or paresis). PI associated with SCI are considered permanent, continuous, and static, with the severity of the PI ranging from mild to severe.

Tremors

A tremor is an involuntary movement (Hallett, 1991). Tremors are produced by involuntary muscle activity and are often described as rhythmic shaking movements. Some tremors occur when muscles are in use (*action tremors*) while others when the muscles are resting (*resting tremors*). *Intention tremors* occur when an individual makes a purposeful motion. Tremors are a common consequence of various health conditions (e.g., MS or stroke). In these situations, the tremors are described under the appropriate health condition (e.g., see MS or stroke).

Essential tremors usually begin in early adulthood and slowly become more obvious (essential tremors that begin in older adults are referred to as *senile tremors*). Essential tremors typically stop when the arms (or legs) are at rest, but become more obvious when the arms are held away from the body or in awkward positions. Essential tremors are relatively fast but result in little movement. The vocal cords can be affected by essential tremors, resulting in inconsistent speech. Essential tremors are permanent, intermittent, and can be progressive. Essential tremors are typically mild initially, but may progress to be severe and can be variable. While tremors are technically a movement PI, essential tremors have no known cause, which is why they are included in this list of health conditions.

TABLE 42.1 Relationship Between the Level of a Spinal Cord Injury and the Resulting Effects on Muscles

Level of Injury	Effect
C1 to C5	Paralysis of muscles used for breathing, controlling the arms, and controlling the legs.
C5 or C6	Paralysis of the legs. Some ability to flex the arms remains.
C6 or C7	Paralysis of the legs. Paralysis of part of the wrists and hands. Much of the ability to move the shoulders and to bend the elbows remains.
C8 or T1	Legs and trunk paralyzed. Hands paralyzed. Arms remain relatively unaffected.

TABLE 42.2 Residual Motor and Sensory Function Associated with each ASIA Score

ASIA Score/ Completeness	Description of Motor and Sensory Function
A	Complete. No residual motor or sensory function below the level of the injury.
B	Motor complete, sensory incomplete. There is no residual motor function, but some sensory capabilities remain in tact.
C	Motor and sensory incomplete. Most muscles below the level of injury are below 60% of normal strength.
D	Motor and sensory incomplete. Most muscles below the level of injury are at or above 60% of normal strength.
E	Normal motor and sensory function.

Locked-In Syndrome

Locked-in syndrome is a rare condition in which a person is conscious and able to think but is so severely paralyzed that communication is possible only by opening and closing the eyes in response to questions (Berkow, 1997). Many diseases or injuries may result in locked-in syndrome, including ALS, subcortical hemorrhage, polyneuropathy, Guillian-Barré syndrome, and brain injuries.

Many cases of locked-in syndrome are the result of various progressive health conditions, such as ALS. In these cases, individuals gradually lose their physical capabilities, usually starting from the lower limbs and progressing to the body and arms. Eventually, speech and movement of the neck are affected. Locked-in syndrome results in muscle power and movement PI including paralysis of most muscles, stiff muscles, and tremors. The PI are permanent, continuous, and severe. The initial severity may be less severe depending on the underlying disease or injury, but by the time an individual is considered locked-in, muscle power PI are always severe.

Summary

While the examples listed are some of the most common health conditions that result in PI, this is by no means a complete list. Table 42.3 provides a summary of the information presented earlier, including both the PI and any additional impairments that associated with each health condition that may influence the design of alternative strategies or technologies.

Associated PI are listed as well as additional impairments associated with the health condition that could interfere with the use of standard computing technologies.

SITUATIONALLY-INDUCED IMPAIRMENTS AND DISABILITIES

As Fig. 42.1 indicates, both the environment in which an individual is working and the activities in which she is engaged can contribute to the existence of impairments that interfere with the use of computing technologies. Environmental characteristics such as lighting, noise, vibration, and temperature can all affect an individual's ability to interact with computing technologies. Similarly, the user may be engaged in multiple activities, each placing demands on her cognitive, perceptual, or physical capabilities, resulting in an overload such that one or more of the tasks cannot be completed optimally. Stress can also affect an individual's ability to interact with computers.

In some situations, an individual's context creates temporary impairments. In other situations, there is no impairment, but a temporary disability does exist. When the individual's physical capabilities are altered, the result is a temporary PI. For example, extended exposure to cold temperatures can make it difficult for an individual to bend his fingers, creating a temporary mobility impairment (similar to arthritis) that can hinder the use of information technologies. In contrast, when the individual's physical capabilities are not altered (e.g., a vibrating environment), but are simply rendered inadequate for the task, the result is a temporary disability. An example is an environment that is vibrating (e.g., a moving vehicle). Vibration does not alter any of the body's structures or functions so there is no impairment, but can make it difficult to use a stylus to enter text using gesture recognition, so the individual does experience a disability. Whether the context results in an impairment or disability, our goal is to minimize the negative impact on the user's computing activity.

While extensive research has studied dual-task scenarios, in which participants were required to attend to two tasks at the same time, these studies tend to highlight decreases in cognitive performance rather than in physical performance. At the same time, individuals are finding themselves in situations in which they are engaged in secondary computing tasks with increasing frequency. For example, a paramedic's hands may be busy providing medical care while they engage in a secondary computing activity to take notes and complete required forms. Simply walking can also make the physical activities required to interact with a mobile computing device more complex (e.g., Price, Lin, Feng, Goldman, Sears, & Jacko, 2006).

While many factors can adversely affect an individual's ability to perform the physical actions required to interact with computing technologies, and some studies have been conducted, these relationships have yet to be comprehensively investigated and reported within the HCI literature. At the same time, the fundamental characteristics of these situationally induced impairments and disabilities (SIID) are clear. Most importantly, unlike impairments associated with health conditions which tend to be permanent, SIID are temporary. As a result, individuals experiencing SIID are less likely to develop accommodation strategies and are therefore more likely to benefit from technology-based accommodations. Since SIID are temporary, they are also intermittent. SIID may be progressive, regressive, or static depending on the conditions and can range from mild to severe or could be variable. Effectively addressing the highly variable nature of SIID is an interesting, and challenging, area in need of additional research.

HCI RESEARCH AND PI

Researchers have approached the issue of designing computing technologies for individuals with PI from a variety of perspectives. Many projects were clearly motivated by a desire to address the needs of individuals with disabilities, and a subset of these projects focus on specific health conditions or PI. For example, researchers have investigated the development of a new interaction alternative for individuals with cerebral palsy (Roy, Panayi, Erenshteyn, Foulds, & Fawcus, 1994), cursor control for individuals with spinal cord injuries (Casali, 1992), and text entry for individuals with spinal cord injuries (Sears, Karat, Oseitutu, Karimullah, & Feng, 2001). Other projects are motivated by specific impairments or disabilities regardless of the underlying cause. For example, researchers have investigated the effect neck range of motion has on the use of head controls (LoPresti, Brienza, Angelo, Gilbertson, & Sakai, 2000), and the use of force feedback for individuals with various movement and

TABLE 42.3 Common Health Conditions that can Result in PI that Hinder Interactions with Standard Computing Technologies

Health Condition	Common PI and Characteristics	Additional Impairments
ALS	**Muscle power**: Weakness often begins in hands or feet. With time, loss of muscle power spreads and becomes more severe. *Permanent, continuous, progressive, mild progressing to severe (paralysis).* **Movement**: Muscles can become stiff. Spasms and tremors can occur. *Permanent, continuous, progressive, mild to severe.*	In time, muscles throughout the body can be affected, including those involved in producing speech.
Arthritis	Osteoarthritis **Mobility**: Reduced range of motion, weakness, difficulty with flexibility of joints and movement. *Permanent, intermittent or continuous, progressive, mild to severe.* Rheumatoid arthritis **Mobility**: Reduced range of motion, weakness, difficulty with flexibility of joints and movement. *Permanent, continuous, progressive, mild to severe.*	None. None.
Brain injury	**Movement**: A wide range of movement-oriented difficulties are possible depending on the location and severity of the injury. *Permanent, continuous, static, mild to severe.*	Speech, vision, and cognitive impairments are possible.
Cerebral palsy	Spastic **Muscle power**: Weakness. *Permanent, continuous, static, mild to severe. Paralysis does not occur.* **Movement**: Stiff muscles. *Permanent, continuous, static, mild to severe.* Choreoathetoid **Movement**: Spontaneous, slow, uncontrolled muscle movements as well as jerky, abrupt movements. *Permanent, continuous, static, mild to severe.* Ataxic **Muscle power**: Weakness. *Permanent, continuous, static, mild to severe. Paralysis does not occur.* **Movement**: Poor coordination, trembling, difficulty with rapid or fine movements. *Permanent, continuous, static, mild to severe.* Mixed Any combination of two forms listed earlier.	Speech and intelligence can be affected. Speech and intelligence can be affected. Speech and intelligence can be affected.
Locked-in syndrome	**Muscle power**: Locked-in syndrome involves complete loss of power in all muscles except those that control the eyes. *Permanent, static, severe.* **Movement**: Muscles can become stiff. Spasms and tremors can occur. *Permanent, static, severe.*	Speech is affected.
Missing limbs or digits	**Mobility**. *Permanent, continuous, static, mild to severe.*	None.
Multiple sclerosis	**Muscle power**: Weakness. *Permanent, intermittent, progressive, mild to severe. Paralysis does not occur.* **Movement**: Clumsiness, tremors, stiff muscles. *Permanent, intermittent, progressive, mild to severe.*	Vision can be affected.
Muscular dystrophy	**Muscle power**: Weakness that spreads and becomes more severe. *Permanent, continuous, progressive, mild to severe. Paralysis does not occur.* **Mobility**: Joints that cannot be fully extended. *Permanent, continuous, progressive, mild to severe.*	None.
Parkinson's disease	**Movement**: Tremors (pill rolling) of one side are often the first sign. *Permanent or temporary; continuous or intermittent; progressive, regressive, or static; mild to severe.* Difficulty initiating movements and stiff muscles. *Permanent, continuous, progressive, mild to severe.* **Mobility**: Difficulty bending the elbow. *Permanent, continuous, progressive, non-existent to severe.*	Speech can be affected. Cognitive impairments are possible.
Repetitive stress injuries	**Mobility**: Reduced range of motion, weakness, difficulty with repetitive motions. *Permanent, continuous or intermittent, progressive, mild to severe.*	None.
Stroke	**Movement**: A wide range of movement-oriented difficulties are possible depending on the location (in the brain) and severity of the injury. Stroke does not always result in movement disorders. *Permanent, continuous, static, mild to severe.*	Speech, vision, and cognitive impairments are possible.
Spinal cord injuries	**Muscle power**: The specific muscles affected are determined by the level of the injury. How much power is lost is determined by the completeness of the injury. *Permanent, continuous, static, mild to severe.*	None.
Tremors	**Movement**: Tremors often affect the hands. *Permanent, intermittent, progressive or static, mild to severe or variable.*	Speech can be affected.

muscle power PI that affect the hands (Keates, Langdon, Clarkson, & Robinson, 2000). Numerous articles that investigate the needs of individuals with specific health conditions or PIs are discussed later.

Another group of projects address issues associated with PI, but do not identify any specific PI or health conditions as the motivation for the research. For example, several articles discuss communication issues for individuals "with severe speech and motor impairments" (McCoy, Demasco, Jones, Pennington, Vanderheyden, & Zickus, 1994), while others explore speech-based interfaces for "motor-control challenged computer users" (Manaris & Harkreader, 1998), new techniques for eye tracking that "could be useful to some people with physical disabilities" (Patmore & Knapp, 1998), and the use of word-prediction for text entry by "people with severe physical disabilities" (Garay-Vitoria & González-Abascal, 1997). Multiple articles that fit into this category will also be discussed.

The goals for this literature review include (a) summarizing the current state of our knowledge with respect to designing computing technologies for individuals with PI, (b) highlighting numerous unsolved problems in need of additional research, and (c) providing pointers to related articles that are not discussed. While there are many ways to organize this review, we group articles by health condition or PI whenever possible. Articles that focus on a specific health condition or PI are discussed, followed by articles that address less-precisely defined PI (e.g., various impairments that affect the hands, regardless of the underlying health condition). Next, we present several articles that have a clear association with PI, but do not discuss any specific PI (e.g., articles that present a technology specifically in the context of individuals with PI, but do not identify the PI or health condition). We conclude by reviewing articles that explore the adoption or abandonment of assistive technologies. Occasionally, articles that focus on a specific health condition are included in other sections due to the PI or technology discussed (e.g., Doherty, Cockton, Bloor, & Benigno, 2000). Throughout this section, we use terminology that is consistent with the definitions presented earlier. As a result, our terminology may differ from that used in the original articles.

Before discussing specific articles, we must acknowledge that several related bodies of literature are not discussed. First, we do not discuss the articles that simply mention individuals with disabilities as one of the possible groups of users who may benefit from the results of the research. For example, many articles discussing multimodal interfaces mention potential benefits for individuals with disabilities at some point, but few are designed for or evaluated using individuals with disabilities (see chapter 21 for an introduction to multimodal interactions and pointers to the relevant research). Second, we do not discuss articles that focus on the more general problem of designing accessible computer systems without a significant focus on PI. For articles of this nature, the reader is referred to the proceedings of the ASSETS conferences (ASSETS, 2006), CUU conference (CUU, 2003), and the ERCIM Workshops on User Interfaces for All (ERCIM, 2006). Stephanidis (2001) also discussed these issues. We encourage the reader to review this literature as it often provides valuable insights into the methods that can be used to more effectively integrate accessibility concerns into the development process. Finally, we do not discuss the extensive literature available in the rehabilitation and assistive-technology communities. We do encourage the reader to review the contributions of these communities. More specifically, we refer the reader to the proceedings of the annual Closing the Gap conferences (CTG, 2006), Rehabilitation Engineering and Assistive Technology Society of North America conferences (RESNA, 2006), and the CSUN conferences (CSUN, 2006). The *IEEE Transactions on Rehabilitation Engineering* often includes articles discussing the technical details of new devices and various books and Internet sites can provide valuable insights (e.g., Edwards, 1995; Gray, Quatrano, & Liberman, 1998; King, 1999; Stephanidis, 2001; Trace Center, 2007).

Spinal Cord Injuries (SCI)

SCI are one of the most frequently studied health conditions in the HCI community. Given the relationship between the level of injury and the resulting impairments, studies typically include individuals with injuries that are no lower than C7. The following articles discuss various topics including cursor control, validating keystroke-level models, a brain-controlled interface, and the use of speech recognition for communication-oriented activities.

Casali (1992) investigated the efficacy of five cursor-control devices when used by individuals with SCI. Twenty individuals with SCI, and ten individuals with no PI, participated in this study. Participants with SCI were divided into two groups using a custom assessment test (see Casali & Chase, 1993; Casali, 1995) that evaluates upper-extremity motor skills. As a result, three groups of participants were discussed: low motor skills, high motor skills, and participants with no PI. Each participant used a trackball, mouse, tablet, and joystick in addition to the cursor keys to complete a series of tasks that required either selecting or dragging a target. Participants in the low-motor-skill group took longer to complete the tasks than the individuals with no PI, but the difference between participants in the high-motor-skill group and the participants with no PI was not significant. For all three groups, the mouse, trackball, and tablet resulted in the shortest task-completion times, the cursor keys took longer, and the joystick was the slowest device. Target size was more important as motor skills decrease. Interestingly, the authors noted that all participants were able to complete the tasks with all five devices with minimal customization. However, individuals in the low- and high-motor-skills groups experienced substantial difficulty when using the mouse, especially when tasks required holding a button down while moving the mouse.

Koester and Levine (1994) investigated the efficacy of keystroke-level models for predicting performance improvements which individuals with SCI would experience when using word prediction software. The authors described a study that provides insights into the benefits word prediction software can provide and the effectiveness of keystroke-level models for predicting performance by individuals with PI. Eight individuals with no PI and six individuals with high-level SCI participated in the study. The individuals with SCI used their normal method of interacting with a keyboard (two used a mouthstick, four used hand splints). Participants completed the tasks with and without word prediction to allow performance improvements to be evaluated.

Two models were explored: (a) a generic model based upon parameters derived from the existing literature combined with new data from individuals with no PI, and (b) a user-driven model based upon parameters derived from the data collected in the current study. The user-driven models resulted in an average error rate of 6% while the generic models resulted in error rates of 53% and 11% for the participants with and without PI, respectively. Perhaps more interesting were the changes in data entry rates observed for the two groups of participants. Overall, the models predicted modest improvements in performance ranging from slight decreases to improvements of over 40%. Results for participants without PI matched the predictions, but participants with PI showed a consistent, and large, decrease in data-entry rates when using word prediction (ranging from approximately 20–50% reductions in data-entry rates).

Mason, Bozorgzadeh, and Birch (2000) and Birch, Bozorgzadeh, and Mason (2002) described a brain-controlled switch that can be activated by imagining movement. Earlier articles described studies in which this technology was used by individuals without PI (Lisogurski & Birch, 1998; Bozorgzadeh, Birch, & Mason, 2000), demonstrating that this technology could effectively identify both real and imagined finger movements. In the current articles, the authors describe a study that involved two individuals with high-level SCI with no residual motor or sensory function in their hands. Their results indicated false-positive rates of less than 1% (e.g., the system detected activity when none was intended) and hit-rates of 35–48% (e.g., the system detected activity when it was intended). While higher hit-rates are desirable, these results were viewed as positive in that they confirmed that individuals with no residual motor or sensory function in their hands could activate the system by imagining finger movements.

Sears, Karat, Oseitutu, Karimullah, and Feng (2001) discussed the results of an experiment designed to explore the effectiveness of speech recognition for communication-oriented activities when used by individuals with SCI as well as individuals with no PI. The individuals with SCI all had injuries at or above C6 with ASIA scores of A or B, resulting in either no use or limited use of their arms and their hands being paralyzed. Overall, there were no significant differences between the two groups of users, with participants producing text at approximately 13 words per minute (wpm) with few errors. An analysis of subjective satisfaction ratings indicated that participants with PI were more positive about their experience. A more detailed analysis revealed significant differences with respect to the processes employed by the two groups of participants. For example, participants with PI spent more of their time dictating and interrupted their dictation more often to correct errors, but they also spent less of their time navigating from one location to another within the document. Oseitutu, Feng, Sears, and Karat (2001) provided additional details that highlighted the different strategies adopted by these two groups of participants.

Sears, Feng, Oseitutu, and Karat (2003) focused on speech-based navigation techniques to help people with spinal cord injuries edit text documents. They found that traditional target-based techniques (e.g., "Select book") and direction-based techniques (e.g., "Move up four lines") were highly ineffective with users spending one third of their time issuing navigation commands while experiencing failure rates of approximately 15% to 19% respectively. Further, when navigation commands failed, the consequences were often rather severe. Frequently, failed commands would alter the content of the document, requiring significant time and effort to recover. The authors proposed several changes to existing navigation techniques to address these difficulties. These revisions significantly reduced the time spent on navigation and significantly improved data-entry rates. Feng and Sears (2004) proposed a new speech-based navigation solution, which introduces carefully specified navigation anchors to allow users to select target words efficiently and reliably. A preliminary study confirmed that their new anchor-based navigation technique allowed users to reach targets quickly while experiencing fewer errors.

LoPresti and Brienza (2004) investigated use of head-controlled devices by individuals with SCI. Although head-controlled devices can help people with limited use of their hands, the devices can prove challenging when the users also have limited neck movement. To address this issue, the authors developed software that automatically adjusted the interface sensitivity to the needs of a particular user. The software was evaluated in two stages. The first stage involved 16 novice users with SCI or multiple sclerosis. Icon selection speed increased significantly with the new software. The second stage involved five users with extended use of the original head-controlled device. Only one participant experienced improvements in performance. Unfortunately, calibrating their system required five steps, resulting in a complex and time-consuming process, especially for novice users who have yet to commit to long term use of the system.

Cerebral Palsy (CP)

CP can result in a variety of impairments due to reduced muscle power or a loss of control over voluntary or involuntary movements. CP can also result in both impaired speech and intelligence. As a result, designing technologies for individuals with CP can be particularly challenging. In this section, we discuss the results of a single study that was motivated explicitly by CP. Additional studies discussed in elsewhere in this chapter may have been motivated by CP. For example, Alm, Todman, Elder, and Newell (1993) described a system designed for individuals with severe speech and motor impairments, including an evaluation that included one individual with CP.

Roy et al. (1994) discussed the design of a gestural interface designed for individuals with speech and motor impairments due to CP. Fourteen students with CP, ages 5–17, were observed during their regular school activities. Numerous communicative acts were observed that combined facial expressions; eye-gaze; vocalization; dysarthcic (slurred, slow, difficult to produce, and difficult to understand) speech; and upper extremity gestures using the head, arms, hands, and upper torso. Subsequently, four students participated in sessions in which data were collected for a predefined set of gestures (using a 3-D magnetic tracker and EMG electrodes). An evaluation of the complete set of gestures, or the subsequent use of such gestures by individuals with CP, was not reported. However, preliminary results that were reported provided encouragement with respect to the use of computer-based gesture recognition as a communications tool for individuals with CP.

Impairments of the Hands and Arms

Multiple studies have investigated the difficulties individuals experience due to movement or muscle power PI that affect their hands and arms. These studies often include participants with a range of underlying health conditions (e.g., CP, SCI, stroke) as well as varying PI. The unifying theme is that each study focused on the ability of the participants to interact with computers using the arms and hands.

Trewin and Pain (1999) discussed the errors that occur when individuals with PI that affect their hands and arms interact with a keyboard and mouse. Twenty individuals with various PI participated, as did six individuals with no PI. Each participant completed tasks using the keyboard, mouse, or both. Several important categories of typing errors were identified: long key presses, additional keys (both local and remote), missing keys, simultaneous keys, and bounces. Each type of error occurred more often for the participants with PI. For additional related work, see Trewin (1996) as well as Trewin and Pain (1998a, 1998b).

Keates et al. (2000) discussed the use of haptic feedback for individuals with PI. More specifically, they investigated the potential of force feedback to facilitate point-and-click activities. They reported on two pilot studies that focused on individuals with movement and muscle power PI but unaffected sensitivity of touch. Six individuals participated in the studies; four participants had choreoathetoid CP, one had Friedrich's Ataxia, and one had Kalman-Lamming's Syndrome. Participants completed a series of point-and-click activities. In the first study, the efficacy of several alternative forms of feedback was explored. These forms included a pointer trail, changing the color of a target when the cursor is over it, force feedback gravity wells that pulled the cursor toward targets, vibrating the mouse when the cursor is over a target, or a combination of all of these forms of feedback. Their results indicated that gravity wells may reduce target selection times by 10–20%. Similar benefits were also seen for pointer trails—a standard accessibility option in the Windows operating systems. Participants disliked the vibrating cursor, which also increased target selection times. Interestingly, these results are similar to those reported for participants with no PI using the Phantom input device (Oakley, McGee, Brewster, & Gray, 2000).

The researchers' second study explored the relationship between force feedback and target size. Force feedback reduced the time required to select targets by 30–50% and error rates by approximately 80%. Without force feedback, the time required increased as target size decreased, but this pattern was greatly reduced when force feedback was provided. Interestingly, at least one participant was unable to complete the tasks without force feedback, but successfully selected the targets when force feedback was provided. These results suggest that force feedback could prove useful for individuals with PI, but it should be noted that these results may be unique to the specific form of force feedback utilized in the current study. Careful evaluation of other forms of force feedback is still necessary.

Building on the previous study, Hwang, Keates, Langdon, and Clarkson (2003) investigated the use of force feedback gravity wells for point-and-click tasks when an undesired haptic distracter is present on the screen. Ten individuals with PI together with eight individuals without PI participated. Their results indicated that the location of the distracter has an important effect on performance. While selection times and error rates improved when force-feedback gravity wells were present, participants were significantly slower when the distracter was directly between the start point and the target. The greatest improvements were evident for the most impaired users.

Keates, Clarkson, and Robinson (2000) investigated the use of the Model Human Processor (Card, Moran, & Newell, 1983) for describing the behavior of computer users with PI. This model combines perceptual, cognitive, and motor activities to predict the time required to complete a task, but was originally developed with an emphasis on individuals without PI. Two groups of individuals took part in their study. The first group consisted of six individuals with PI caused by various health conditions (e.g., quadraplegia, MD, spastic CP, choreoathetoid CP, and Friedrich's Ataxia) while the second group included three individuals with no PI. Their results are summarized in Table 42.4. [AQ13]

Participants with PI appeared to take longer to complete fundamental perceptual and cognitive activities. These results are consistent with those discussed earlier (Koester & Levine, 1994), but should be interpreted carefully given the varied health conditions of the participants. Through observation and an analysis of the different times recorded for motor activities, the authors concluded that additional cognitive or perceptual activities were taking place when individuals with PI completed basic motor tasks (as compared to individuals with no PI). While their results are not definitive, they confirm the need for additional research that focuses on the time required to complete basic perceptual or cognitive activities.

Keates, Hwang, Langdon, and Clarkson (2002) investigated the challenges of the point-and-click tasks for people with PI by studying detailed records of cursor movements. The motivation for their study was the assumption that traditional task measures such as speed and error rate are not sufficient if the goal is to fully understand the interaction process including the reasons users experience difficulties. Five individuals with PI and three without PI completed a series of multidirectional cursor positioning tasks, providing detailed information regarding the path the cursor followed during each activity. Based on those data, six metrics including the distance traveled relative to cursor displacement, distribution of distance traveled for a range of cursor speeds, submovements, distance traveled away from the target, distribution of distance traveled for a range of curvatures, distribution of distance traveled for a ranged of radii from the target, were proposed which focused on deviations from the optimal path for a number of predefined cursor positioning tasks. Next, a subset of the participants (four with PI and

TABLE 42.4. Average Time Required, By Individuals With and Without PI, to Complete Fundamental Perceptual, Cognitive, and Motor Tasks

	Perceptual	Cognitive	Motor
No PI	80 ms	93 ms	7 0ms
PI	100 ms	110 ms	110 ms, 210 ms, 300 ms

three without PI) completed a set of target selection tasks. These tasks differed slightly from those used in the initial study in that a circular starting point was defined near the center of the screen. The results indicated that the new metrics helped highlight differences between users with and without PI, especially those difficulties experienced by the more impaired users.

Building on the results of the previous study, Hwang, Keates, Langdon, and Clarkson (2004) investigated the submovements of the cursor path for individuals with PI. When conducting navigation tasks, users completed a rapid movement that is comprised of a sequence of smaller, discrete submovements. The authors studied the cursor trajectories of six users with PI together with three individuals without PI. The results showed that some individuals with PI paused more often and longer than participants without PI, resulting in up to five times more submovements to complete the same task. There was a correlation between error rate and peak submovement speed for users with PI, which was not observed for users without PI.

Myers, Wobbrock, Yang, Yeung, Nichols, and Miller (2002) focused on individuals with PI that result in a loss of gross motor control, such as the movement of the arm and the wrist, but retention of specific fine motor control, such as the finger. Software was developed allowing the use of a PDA as the mouse and keyboard. Four participants with muscular dystrophy tested the software; all provided positive feedback. The parents of one participant noted that although performance was slower using the PDA as compared to the regular keyboard and mouse, the PDA was much less tiring and could be used much longer without a break.

Impairments in Infants

Infants with severe PI often develop to be passive with limited or nonexistent speech. This is true even when cognitive skills are normal. Fell, Delta, Peterson, Ferrier, Mooraj, and Valleau (1994) discussed the development and evaluation of the Baby-Babble-Blanket as a technology to enable infants with severe PI to control their environment and communicate. One specific goal was to help these infants improve their motor skills. The key component of the system is a pad, containing 12 equally spaced switches, which can be placed on the floor. When a switch is pressed, a computer provides feedback. The feedback could include digitized speech, music, and sounds. The authors report on a single-subject experiment involving a five-month-old infant with poor muscle tone, hydrocephaly, and clubfeet. This infant was able to activate switches, became more active when sounds were played in response to switch activations, associated switches with a desired effect (e.g., hearing his mother's voice), and was able to modify these associations when the feedback was moved from one switch to another.

Significant Speech and Physical Impairments (SSPI)

While the terminology may vary, several studies have explored technologies for individuals with significant speech and physical impairments. The article by Roy, Panayi, Erenshteyn, Foulds and Fawcus (1994) discussed earlier could have been included here, but was listed separately due to its explicit focus on impairments caused by CP.

Alm, Todman, Elder and Newell (1993) described the development of a computer-aided communication technology for individuals with SSPI. This article focused on situations in which the speech impairment results from PI rather than cognitive impairments. The system tracks the conversation, guiding users as they specify the type of utterance they wish to generate. The system was evaluated with two participants. The mean rate of speech was 144.4 word per minute (wpm). One participant with no PI took part in eight conversations with student volunteers, producing speech at a rate of 67.4 wpm while the conversation partners spoke 132.9wpm. While selecting an utterance could require one to four clicks, less than 0.5% of the selections involved more than two clicks. This suggests that the desired utterances were easily accessed given the current design of the system. The second participant had CP and used a word board containing 400 words as his primary method for communicating. Using the experimental system, this participant was able to express himself more fully, increasing his vocabulary from 143 to 534 words and the number of times he took control of the conversation from 10 to 27. Many important questions must still be addressed for systems such as the one described in this article. One critical issue is the effectiveness of such a system as the number of conversation topics and phrases increases. Another critical issue is how the user can efficiently enter and organize these phrases as they customize the system for use in new situations.

Demasco, Newell, and Arnott (1994) discussed the development of a system to support communication activities based upon principles of visual information seeking. Building on five key principles derived from the literature, a computerized word board was designed. Key features included the separation of navigation and selection which allows these activities to be supported by different devices, and expanding the collection of accessible words using word associations (e.g., is-like, isa, hasa, goes-with). At the time the article was written, the system was under development and a variety of directions for research were presented, but no evaluation had been completed.

McCoy, Demasco, Pennington, and Badman (1997) investigated the application of natural language processing to Augmentative and Alternative Communication (AAC) to improve the communication capability for individuals who have cognitive disabilities that hinder their ability to speak. The objective is to develop intelligent communication aids that provide linguistically "correct" output when speeding the communication rate. The author proposed a prototype system with a key component called the "Intelligent Parser Generator." Several research issues related to the system and the specific user population were proposed and discussed. However, no user evaluation result has been reported.

Albacete, Chang, Polese, and Baker (1994) and Albacete, Chang, and Polese (1998) discussed the design of iconic languages for use by individuals with SSPI. Their approach is based upon semantic compaction in which ambiguous icons are combined to represent concepts. For example, "APPLE VERB" could mean "eat" while "APPLE NOUN" means "food." An icon algebra was defined, including operators to combine, mark, provide

context for, enhance, or invert icons. As a result, this approach can allow a vocabulary of several thousand words to be represented with just 50 to 120 icons. This approach served as the foundation for the Minispeak system.

Input Using Electrophysiological Data

Traditional input techniques involve some kind of physical activity. Most often, the hands and arms are involved, but other technologies allow users to provide input by moving their eyes, heads, tongue, or eyelids. However, a growing number of researchers are investigating input techniques that do not require physical activities. The article by Mason, Bozorgzadeh and Birch (2000) discussed earlier provides one example, in which electrophysiological data were used to generate input in a computer. In this section, several additional articles that discuss the use of electrophysiological data are discussed. This is a new, exciting, and promising form of input for individuals with significant PI.

Patmore and Knapp (1998) discussed an approach to eye tracking that uses the electrooculogram (EOG) and visual evoked potentials (VEP) as input. While a number of EOG-based eye-tracking systems have been discussed, additional research is needed to make such systems reliable (see LaCourse & Hludik, 1990; Patmore & Knapp, 1995). The VEP is a response to a flash of light that is highly sensitive to the stimulus' distance from the center of the field of vision. The authors presented the technical details of a system that combines the EOG, the VEP, and fuzzy logic to improve accuracy. While promising, significant additional research is necessary before such a system will be sufficiently reliable for use under realistic conditions.

Allanson, Rodden, and Mariani (1999) described a toolkit designed to explore the use of electrophysiological data to support interactions with computers. They discussed the variety of electrophysiological data available including the electroencephalograph (EEG), electromyograph (EMG), and galvanic skin resistance (GSR). The EEG is an electrical trace of brain activity, the EMG can detect the state (e.g., complete relaxation, partial contraction, complete contraction) of a muscle, and the GSR measures changes in the resistance of the skin. The authors provided interesting references describing the use of the EEG to turn switches on and off (Kirkup, Searle, Craig, McIsaac, & Moses, 1997) and to differentiate between five tasks (Keirn & Aunon, 1990), as well as references illustrating the use of the EMG to control computer games (Bowman, 1997) and manipulate objects on a computer (Lusted & Knapp, 1996). Their toolkit integrated these signals using predefined widgets that allow the signals to be used to drive applications.

Barreto, Scargle, and Adjouadi (1999) discussed the design of an integrated assistive real-time system as an alternative input device to computers for individuals with significant PI. The system uses EMG biosignals from cranial muscles and EEG from the cerebrum's occipital lobe to control mouse input. The system, combined with an onscreen keyboard, provides full operation functionalities with computers. Six healthy participants tested the system. Participants required an average of 13–21 seconds to select a button. Although the selection speed is quite low, the study demonstrated that the EMG and EEG based system could allow users to complete target selection tasks.

Kübler, Kochoubey, Hinterberger, Ghanayim, Perelmouter, Schauer, Fritsch, Taub, and Birbaumer (1999) discussed the use of slow cortical potentials in a 2-s rhythm to control cursor movements on a computer screen. Three individuals with advanced ALS participated in extensive practice sessions to learn to control this signal. Ultimately, two of the three reached accuracy rates between 70 and 80%, which allowed these individuals to select letters and words displayed on a computer screen. Building on these results, Birbaumer, Hinterberger, Kübler, and Neumann (2003) developed a Thought-Translation Device (TTD) using slow cortical brain potentials. The system allows the user to directly select letters and words with brain response alone. Eleven locked-in patients tested the device with a training period ranging from six months to six years. Nine participants had end-stage ALS, one had experienced a subcortical hemorrhage, and one had polyneuropathy and Guillian-Barré Syndrome. At the beginning of the training session, the seven participants who were not completely locked-in patients were able to communicate "yes" or "no" signals with face or eye muscles. Two of these seven patients lost all motor control during the study but were still able to communicate using TTD. None of the patients who were completely locked-in at the beginning of training were able to gain effective control over the system, but one patient did reach a selection speed of one letter or word per minute. Although only two participants were able to use this system successfully, this study provided additional confirmation that a brain-computer interface has the potential to benefit users with severe PI.

Craig, Tran, McIsaac, Moses, Kirkup, and Searle (2000) developed a system that employs Alpha waves, which increase in amplitude when the eyes are closed and decrease in amplitude when the eyes are open, to activate electrical devices. Ten individuals without PI and one individual with severe PI evaluated the system. The participant with PI had childhood-onset poliomyelitis which resulted in the loss of motor function below her neck. Participants without PI completed one 50–60 minute session on each of four consecutive days. In each session, they activated the environmental control system 30 times. The participant with severe PI completed four trials of approximately ten minutes duration in each trial. There was a significant decrease in selection times between day one and all subsequent days, including a 16% reduction from 12.7 seconds on day one to 10.7 seconds on day two. Error rates also decreased significantly from approximately 50% to 23% by the second day.

Doherty et al. (2000) discussed the use of formative experiments and contextual design to assess the potential of applications controlled by eye movements, the EOG signal, and the EEG signal (e.g., the Cyberlink interface). An initial study involving 44 participants with various PI demonstrated some expected limitations (e.g., the mouse was not effective when arm/hand control was limited, eye tracking was not effective when peripheral vision was impaired) as well as unexpected difficulties (e.g., some quadriplegic participants could not produce signals below the neck and therefore could not operate the GSR device). Interestingly, the Cyberlink interface was the only alternative that all 44 participants could use to navigate a maze. Based on those results, Doherty et al. (2001) proposed ten design guidelines for developing interfaces with Cyberlink and

brain-injured users. Example guidelines include avoiding strobes when sampling the cursor, using tunnels instead of clicking for selection, avoiding long distances by providing rest stops, and not using timeouts to indicate selection. Overall, their results confirmed that interfaces built upon electro physiological data can be useful for individuals with PI, but that substantial research is necessary before the full potential of these technologies will be realized.

Gnanayutham, Bloor, and Cockton (2005) improved the CyberLink system described earlier by adding a discrete acceleration function for cursor movement. Two studies were conducted. The first tested the adaptive cursor acceleration algorithm with 30 participants, including 19 with PI. The results suggested that the acceleration function did improve performance, but was not sufficient to overcome the wide variations in capabilities of the study participants. To address this shortcoming, the authors explored personalized tiling for brain interfaces, with the goal of producing a better match between the demands of the device and the capabilities of the users. In the second study, five individuals with brain injuries evaluated the personalized interface. All five participants were able to communicate with the system. Two participants communicated through a two-target, yes/no interface with an average success rate of 70%. Two other participants communicated using a switch, also achieving 70% success rate. Three participants showed interest beyond communication tasks and tried using the interface for television and music systems in their rooms.

Kennedy, Kakay, Moore, Adams, and Goldwaithe (2000) discussed an experimental interface that requires the implantation of special electrodes into specific areas of the brain. For one individual, EMG signals from various muscles were utilized to supplement the neural signals detected by the electrode. With five months of practice, the individual learned to control cursor movements. Results for three tasks were reported. These tasks involved moving the cursor across the screen, placing the cursor on an icon or button, and selecting the icon or button by holding the cursor still for a predefined period (or activating a predefined EMG signal). Through these tasks, this individual was able to invoke synthesized speech for predefined messages, spell names, and answer questions. Moore and Kennedy (2000) identified several directions for additional research including signal understanding, patient training, navigation paradigms, control of neural prosthetics, and communication.

Carroll, Schlag, Kirikci, and Moore (2002) focused on the communications needs of individuals who are locked-in. The authors analyzed log sheets, conducted interviews, and visited users to identify user needs. The style and content of interactions varied according to the role of the person with whom the individual was communicating. Family members engaged in fewer interactions that allowed the individual to participate, while nurses continually questioned the individual about various issues, including their comfort and needs. This finding suggests that the support offered by the system may vary based on the relationship between the individual and her communication partner.

Felzer and Freisleben (2002) developed that "Hands-free" Wheelchair Control System (HaWCoS) that allows individuals to control an electrically powered wheelchair without using their hands. The system uses muscle contractions, a kind of EMG signal associated with any arbitrary muscle, as input signals to communicate with the wheelchair. Three experiments were conducted, comparing one prototype of the HaWCoS system with a traditional joystick control. Although requiring more time than the traditional joystick, the HaWCoS system allowed users to control the wheelchair under a hands-free condition, providing an alternative for people with severe PI.

Borisoff, Mason, Bashashati, and Birch (2004) investigated Low Frequency Asynchronous Switch Design (LF-ASD) as a direct Brain-Computer Interface (BCI) technology for people with severe PI. Earlier research on LF-ASD produced encouraging results, but error rates were too high for practical use. These authors implemented energy normalization and feature space dimensionality reduction features, which normalize the low-frequency signals and increase accuracy. The revised system was evaluated by four individuals with high-level SCI as well as four individuals without PI. Accuracy improved significantly compared to earlier results, with true positive rates increased by approximately 33% and false positive rates by 1% to 2%. Individuals with PI achieved the same accuracy as those without PI, which was not possible when using a traditional keyboard and mouse.

Moore and Dua (2004) developed a biometric input device using galvanic skin response (GSR), which is a change in electric conductivity of the skin caused by increased activity of the sweat glands. Various emotional responses such as fear, excitement, or anxiety can affect the GSR level. A locked-in individual who had lost the ability to use a brain-computer interface tested the system during a one-year period. The results showed that there was a significant difference in accuracy between baseline and controlled states. After several days, an accuracy rate of approximately 62% was achieved. The experiments demonstrated that GSR has the potential to be used by people with severe PI.

Cursor-Control Technologies

Cursor control and text generation are perhaps the two most common activities when interacting with computers. In fact, cursor control can form the foundation for text generation and therefore may be the most fundamental task in which individuals with PI engage. Numerous technologies have been explored for controlling the cursor location and many of the articles summarized earlier focused on cursor control tasks. While new eye-tracking technologies are being explored, commercial systems are readily available and can prove useful for individuals with PI. The issues involved in using these technologies are not discussed in this chapter, but Sibert and Jacob (2000) provided useful pointers to the recent literature on this topic. In the following sections, technologies are discussed that allow individuals to control the cursor using their heads, tongues, eyes and eyelids, or speech.

Head. When an individual has limited use of their hands and arms, other options must be found for interacting with computers. One option is to control the cursor location using head movements. In this section, several articles are discussed that explore the issues involved in interacting with computers via head movements.

Radwin, Vanderheiden, and Lin (1990) discussed the results of a study that compared a standard mouse to a lightweight, ultrasonic, head-controlled pointing device. Ten participants with no PI completed a variety of tasks that required the selection of circular targets. Distance, target size, and device all significantly affected movement times. Interestingly, target size had a greater effect for the head-controlled device, with movement times decreasing more dramatically than when the mouse was used. The direction of the movement also affected movement times, with the horizontal movements resulting in the shortest times for the mouse and vertical movements resulting in the shortest times for the head-controlled device. Results were also reported for two individuals with CP including detailed results for one of these individuals that highlighted the importance of carefully assessing the abilities of individuals with PI when designing computer systems for their use. For example, horizontal movements were efficient for participants with no PI, but the individual with CP experienced more difficulty moving the cursor to the right than any other direction.

Lin, Radwin, and Vanderheiden (1992) reported on a similar study that focused on the relationship between control-display gain and performance with both mouse and head-controlled device. Ten individuals with no PI participated in this study. Movement times and deviation of the cursor from the optimal path were reported. Extensive analyses of the results highlighted the importance of target size, movement distance, and gain. Gain affected movement times, but both target size and movement distance were more influential. A U-shaped curve was observed for both the mouse and head-controlled device. The reader is referred to the original article, as well as Schaab, Radwin, Vanderheiden, and Hansen (1996), for additional information regarding the relationship between control-display gain and performance with head-controlled devices.

Malkewitz (1998) described a system that uses head movements to control cursor location and speech to click the mouse buttons, activate hotkeys, and emulate the keyboard. The goal was to provide access to standard applications by emulating standard input devices (e.g., keyboard and mouse). Speech recognition was restricted to a predefined set of commands and the ability to enter individual letters. Full dictation was not supported for various reasons. While preliminary results were encouraging, the authors note the need for additional research.

Evans, Drew, and Blenkhorn (2000) described the design of a head-operated joystick. The authors briefly summarized the results of an evaluation involving 40 participants including nine individuals with undocumented impairments. It is reported that all 40 individuals used the device successfully, but detailed measures were not provided. The nine individuals with impairments also used a commercially available device that emulated a mouse. Interestingly, all nine preferred the joystick emulation to the mouse emulation provided by the commercial device. It is speculated that this preference is due at least in part to the fact that users can rest their heads in a neutral position once the cursor is placed in the correct location when using a device that emulates a joystick.

LoPresti, Brienza, Angelo, Gilbertson, and Sakai (2000) studied the relationship between neck range of motion and the use of head-controlled pointing devices. Fifteen individuals with no PI as well as ten individuals with various health conditions that resulted in PI that affected neck movements participated in the study. A head-mounted display was used as opposed to a traditional desktop monitor. The results confirmed that the individuals with PI were less accurate and took longer when completing icon selection tasks. They were also less accurate when performing target-tracking tasks. Further, their performance was much more variable than that of the participants with no PI. As with several other studies, the authors found that models developed based upon individuals with no PI do not accurately represent the performance of individuals with PI. A more detailed analysis indicated that vertical movements were faster than horizontal movements and horizontal movements were faster than diagonal movements. LoPresti (2001) reported similar results.

De Silva, Lyons, Kawato, and Tetsutani (2003) developed a vision-based face-tracking system for cursor control. The cursor position was controlled head movements. The mouse button was activated by the individual opening his mouth. Eight users without impairments tested the system. The face tracking system allowed for an index of performance of 2.0 bits/second, which is greater than a joystick, but slower than a trackball and touchpad. The subjective satisfaction measures showed that the system was rated average for ease of use. The lowest ratings were given for smoothness and accuracy of cursor control. The system has the potential to support hands-free typing.

Tongue. Salem and Zhai (1997) provided a brief description of a system that allows the cursor location to be controlled using a tongue-operated isometric joystick like those available on many portable computers. A Trackpoint joystick was mounted in a mouthpiece, similar to those used by athletes, that was fitted to the individual's upper teeth. Two individuals participated in a pilot study. Neither had experience with an isometric pointing device. Both used the tongue-controlled joystick as well as a standard finger-controlled joystick. Initial performance was substantially better when using their fingers, but with practice the gap between the two devices narrowed. During the last trial, the tongue-based device was 5% slower than the finger-based device for one participant, but it was 57% slower for the other participant.

Eye and eyelid. Shaw, Loomis, and Crisman (1995) presented a system designed to allow individuals to control various devices, including computers, by opening and closing their eyes. Two individuals utilized the system. The first (participant A) had a high-level SCI resulting in no residual use of his arms or hands. The second (participant B) had a stroke and could only control a single eyelid. With six hours of practice, participant A was able to move from interacting with simulation software to controlling a powered wheelchair. He found this device to be less obtrusive than the chin-control he normally used to control his wheelchair. After 12 hours of practice, participant B was able to navigate the maze presented by simulation software and made progress toward controlling a powered wheelchair. Given the minimal motor control this individual exhibited, this was considered a noteworthy accomplishment.

Bates and Istance (2002) studied a zooming interface to enhance the performance of eye-controlled pointing devices. Selecting small targets was challenging due to the way the hu-

man eye functions. The authors addressed this issue by introducing automatic zooming to make targets larger and easier to select. Six individuals without PI took part in the study. The result confirmed that target size is an important factor for the effectiveness of eye-based interaction. The zooming interface was significantly faster and more accurate compared to original solution without zooming. However, the zooming interface was considered less comfortable than the head mouse.

Hornof, Cavender, and Hoselton (2004) developed a system for children with motor impairments to draw pictures and select targets using eye movements. The eye system allowed users to switch between looking and drawing by focusing on a predefined part of the screen. Eight participants without PI evaluated the system. Participants generally found the system easy to learn. The quality of the pictures drawn showed that the system could allow children to draw pictures with their eyes, although it is not an easy task.

Speech. Manaris and Harkreader (1998) and Manaris, Macgyvers, and Lagoudakis (2002) discussed a "listening keyboard" that allowed users with PI to access all of the functionality of a traditional keyboard and mouse. A usability study with three individuals with PI explored the effectiveness of the listening keyboard through a "Wizard-of-Oz" design. Results from a postexperiment questionnaire suggested that participants were comfortable with the concept of a "listening keyboard" and that it would be preferable to mouthstick or handstick systems. The participants were able to complete tasks faster with lower error rates when using the listening keyboard compared to the handstick. However, these results should be interpreted with caution since the number of participants was limited and the "Wizard-of-Oz" methodology results in unrealistically low recognition error rates.

Karimullah and Sears (2002) focused on continuous speech-based navigation techniques and studied a standard speech-based mouse cursor and a predictive cursor. The predictive cursor was proposed to address the delay related to speech processing. As the cursor moves, the predictive cursor appears in front of the actual cursor, indicating the position at which the actual cursor will be located if the command to stop the cursor were issued. Twenty-eight participants without PI evaluated the efficacy of the two solutions. The difficulty users experienced selecting targets was affected by the speed at which the cursor was moving, target size, and the delay between when the command was issued and when it was processed, but no differences were detected between the standard and predictive cursors.

Building on the results of the previous study, Sears, Lin, and Karimullah (2002) conducted an experiment to explore why the predictive cursor did not prove beneficial. This study investigated three sources of delay associated with speech input: the time required for the user to initiate a response, the time required to speak the response, and the time required for the system to produce a response. Fifteen individuals without PI participated in this study. Based on the results of this study, the authors proposed a variable speed cursor to improve the performance as this would allow users to decrease the speed of the cursor when selecting small targets. This is expected to address the problems that users experience and significantly reduce failure rates.

Dai, Goldman, Sears, and Lozier (2004) focused on discrete speech-based navigation techniques, evaluating two grid-based solutions, which allow users to select targets by recursively locating the cursor onto a smaller area of the screen. Both solutions divided the screen into nine regions. One solution provided a single cursor in the center region. The other provided nine cursors, each locating in one region. Twenty-four participants without PI tested the two techniques. The nine-cursor solution allowed users to select targets of varying sizes, distance, and direction significantly faster than the one-cursor solution. Through comparisons with earlier studies, the authors suggested that grid-based solutions might prove advantageous when compared to other speech-based navigation solutions such as the continuous cursor.

Text Entry Technologies

Computers are frequently used to generate text. While the keyboard is the most frequently used alternative, other technologies can also be used. Some individuals with PI select letters from on-screen keyboards, others use speech recognition, and still others interact with the standard keyboard. When using an on-screen keyboard, text entry is effectively turned into a cursor-control task. As a result, all of the cursor-control technologies discussed earlier can also be used for text entry. The article by Sears, Karat, Oseitutu, Karimullah, and Feng (2001) discussed the use of speech recognition for text entry, but was included earlier due to its focus on individuals with SCI. Articles in earlier sections could also be included here, but are discussed separately due to their focus on individuals with specific impairments. The articles included later discuss technologies that can be used for text entry without focusing on any health condition or PI.

Keyboard-based text entry. Matias, MacKenzie, and Buxton (1993) described a one-handed keyboard based upon the traditional QWERTY design. This basic design allows users to access all of the keys on one side of the keyboard at a time, switching between sides by pressing and holding the spacebar. A study with ten participants with no PI was reported. Each participant used his or her nondominant hand when typing with one hand. Speed and errors improved significantly over ten sessions. By the end of ten sessions, participants were typing 34.7wpm with an error rate of 7.44% with the one-handed keyboard as compared to 64.9wpm and 4.20% errors with the two-handed keyboard. While this system could prove useful for individuals with PI that hinder the use of one hand, it is important to remember that the efficacy of this design for individuals with limited experience using the QWERTY keyboard is uncertain.

Many researchers have investigated techniques that allow users to generate text using fewer keystrokes than are normally required. These techniques typically predict words, given just a couple of letters, using statistics that describe how frequently various words are used. Some systems also use statistics regarding how frequently various words follow words that have already been entered. Other techniques present multilevel interfaces in which users repeatedly select categories until the desired word is available. Once the word is selected, an appropriate

ending (e.g., "s," "ing") can be added. Demasco and McCoy (1992) presented a theoretical discussion of a word-based virtual keyboard that uses a level-based interface. McCoy, Demasco, Jones, Pennington, Vanderheyden, and Zickus (1994) discussed a word-based text-entry technique by which users specify uninflected content words and the system adds appropriate endings to the words as well as additional words (e.g., articles, prepositions) to complete the sentence. Garay-Vitoria and González-Abascal (1997) discussed the potential of syntactic analysis of previously entered words to enhance word prediction results. Boissiere and Dours (2000) described a system that predicts word endings without assistance from the user. Unlike many of the articles discussed earlier, all of these articles take either a theoretical or a systems approach to the problem. Consequently, some of the systems have not been implemented and those that have been implemented have not been evaluated by the intended end users.

Trewin (2002) studied overlap errors, one of the common typing errors in which two keys are pressed at once, to improve typing accuracy for people with PI. While some users with PI use keyguards to reduce overlap errors, the author developed a software filter to identify and eliminate overlap errors. The software used keystroke timing and language-based techniques to identify and automatically correct overlap errors. Twenty participants with PI tested the software. The most effective filter was based on keystroke timing characteristics. This filter identified 97% of the errors and misclassified 46% of the deliberate overlaps. The accuracy of the filters was dependent on the typing styles of the users.

Willis, Pain, Trewin, and Clark (2002) developed a system that automatically reconstructs full sentences from abbreviated text, allowing people with PI to type less text. Ten individuals without PI participated in an evaluation study. Four texts were used in the study, each of 500 letters. Each participant completed four abbreviation tasks based on one text and one reconstruction task based on a different text. Both texts were randomly chosen from the four texts. Several abbreviation patterns were identified, such as "double letter" being transformed to single letter and "ch" being transformed to "k." These patterns provided insights that guided the development of an abbreviation expansion system that matches users' natural abbreviation styles.

Trewin, Slobogin, and Power (2002) developed a keyboard optimizer capable of adapting the keyboard to meet the users' needs, with a focus on users with mild to moderate PI that negatively affects their ability to use a keyboard. Given a sample of an individual's typing behaviors, the optimizer generates a customized profile that includes details about how the individual types that can allow the keyboard to be configured to improve typing outcomes for the specific individual. The system was evaluated by eight users with PI that affected their typing. Twenty-one configuration modifications were suggested by the optimizer, with 20 being judged by the clinicians as being appropriate. Participants liked the system and found it easy to use. Trewin (2004) further improved the underlying models and built a dynamic keyboard. Ten individuals with young-onset Parkinson's Disease provided input to the system. The automatic adjustment of the key repeat delay feature was acceptable to the users.

Wobbrock, Myers, Aung, and LoPresti (2004) developed two integrated gestural text-entry methods designed for use from power wheelchairs. Both devices can be integrated into existing controls on power wheelchairs and are small, light, inexpensive, and require minimal configuration. One method uses a joystick and the other a touchpad. Both methods use a stylus-based unistroke text-entry method called EdgeWrite designed for people with tremors. EdgeWrite relies on physical edges and corners to provide stability during motion. The users move stylus, finger, or joystick along the physical edges and into the corners of a square bounding the input area. The system recognizes the input based on the order corners which are hit. Seven power wheelchair users joined the participatory design process and evaluated the two techniques. Although data were too limited for a complete statistical analysis, the results showed potential for the target population. The touchpad application was faster than the joystick application, allowing users to generate an average of one word per minute.

Speech recognition based text entry. Goette (1998) conducted a field study to identify factors that influence the successful adoption of speech recognition software for both dictation-oriented and environmental control tasks. Individuals with various health conditions (e.g., MS, MD, SP, arthritis) participated. Most individuals who stopped using the software (53%) did so within three months. Interestingly, those individuals who successfully adopted the software had higher expectations for the system and expected greater benefits, but also believed the system would be easy to use as compared to those individuals who stopped using the software. When the software was successfully adopted, it was used for a wide variety of computer-based tasks rather than a few isolated activities. Four guidelines were proposed for successful outcomes: (a) managing expectations by understanding the potential benefits and limitations, (b) selecting the correct system for the tasks to be accomplished, (c) obtaining thorough training, and (d) trying the system for an extended period before purchasing it.

Pieper and Kobsa (1999) developed a speech system to allow severely impaired users who are bedridden to interact with the computer. The authors implemented several modifications to the commercially available system to cater the usage environment. The output of the computer was displayed on the ceiling and the speech recognition system configured to accommodate the use of respiratory devices. One patient who had ALS for five years used the system for seven months to write poetry, diaries, and letters. The result shows that the projected video display was acceptable to the user. Overall, the user was satisfied with the system.

Manasse, Hux, and Rankin-Erickson (2000) studied the use of speech recognition to help a 19-year-old survivor of traumatic brain injury (TBI). This study focused on text generation tasks using Dragon NaturallySpeaking. This individual experienced a recognition error rate of approximately 20%. The participant quickly and easily mastered navigation and error-correction commands. Interestingly, the participant generated more text using the standard keyboard than she did using speech recognition, but she generated more complex sentences when using the speech-recognition system. Overall, minimal qualitative differences were observed in the resulting text.

Bain, Basson, and Wald (2002) focused on using speech recognition to generate lecture notes for students in the classrooms. Soft- and hardcopy notes can provide an alternative to traditional note taking for people with PI or perceptual impairments. More recently, Hewitt, Lyon, Britton, and Mellor (2005) developed the "SpeakView" system, which provides speech-to-text displays for meetings and in classrooms. Key issues that were addressed by the "SpeakView" system included automatically adding punctuations, style, and acceptability of the captions, usability, and training. Sixty-nine students evaluated the system by observing two presentations and answering predefined questions. Captions produced by "SpeakView" were provided for one of the presentations. The use of captions did not influence the number of questions answered correctly by the students, but students did prefer presentations with captions. Students indicated that the captions made the talk easier to understand and that they felt the captions helped them answer more questions correctly. This system, if fully implemented with acceptable accuracy, has the potential to benefit individuals with PI or hearing impairments who find taking notes difficult.

Koester (2004) investigated the use of automatic speech-recognition technology by experienced users. Twenty-four users with PI that hindered the use of the keyboard and mouse, with six months of experience with ASR, participated in this study. Each participant completed a series of predefined tasks using two input methods: speech and nonspeech. Participants responded to a survey about their usage patterns. The results showed that speech was significantly slower for command input, but there was no significant difference in text entry rates between speech and nonspeech input. There was a wide variation among text entry rates, ranging from 3–32 words per minute. There was also a wide variation in recognition accuracy, ranging from 72%–94%. The study confirmed that recognition errors are a major limitation for speech-based input, with users spending an average of 56% of their time on error correction. Interestingly, the participants who achieved the best performance were those who tended to employ the prescribed correction strategies.

Eye-gaze-based typing. Lankford (2000) developed an eye-gaze interface that can support both text entry and cursor control tasks. The system used dwell time to provide the user with access to various mouse clicking actions. However, because of the limitations of the human eye, gaze-based clicking was not able to reliably control the Windows environment, particularly for users with PI. To improve the accuracy of gaze-based clicking, a zooming interface was developed to enlarge the objects when needed. Key selection was accomplished by prolonged fixation on the keyboard key. The activation time could be customized to suit individual users. A list of possible words to insert into a document, based on what the user had already typed, was provided to speed up typing and reduce errors. No user study has been reported concerning the effectiveness of this system.

Hansen, Itoh, Tørning, Aoki, and Johanson (2004) investigated the effectiveness of gaze typing compared to hand typing and head tracking. Twenty-seven participants without PI evaluated the three modalities with two interfaces in different languages. Each participant entered predefined text. Participants achieved a typing speed of 6.22 words per minute. The eye-typing system resulted in an error rate of 6% with the Japanese interface and 28% with the Danish interface. Significant learning effects were observed for both interfaces. Overall, gaze typing was the least efficient and least satisfying input modality among the three alternatives. The potential of the gaze-based typing and its advantages compared to other input modalities needs to be investigated by including users with PI, especially those with severe PI for whom gaze typing may be one of the few alternatives that are available.

ADOPTION OR ABANDONMENT OF ASSISTIVE TECHNOLOGIES

It has been estimated that over 1,000 assistive technology products come to the market each year (Scherer & Galvin, 1996). Unfortunately, the adoption rates of assistive technology products are highly disappointing. Many new technologies never reach the majority of the intended users due to a lack of marketing activities or cost. Even when the technologies reach the intended users, many are abandoned within a short period of time. Understanding why users adopt or abandon assistive technologies is critical for the design, implementation, and marketing of these technologies. A number of studies that can provide valuable insights to researchers, designers, as well as healthcare providers are summarized later.

Scherer (1996) studied adoption rates of general assistive technologies and found that many people abandon those technologies. Abandonment rates ranged from 8% for life-support devices to 75% for hearing aids, depending on the type of device. On average, about one third of all assistive technologies were abandoned. However, it was also observed that many people who continue using the technology are unhappy or uncomfortable with the experience, suggesting that the underlying problems are even more significant than the abandonment rates would suggest. It was observed that most abandonment occurs within the first year, especially the first three months, or after five years of use (normally because insurance policies only support the first five years of use). This suggests that users learn relatively quickly whether the specific assistive technology works for them. If not, they abandon it quickly. The author believes that the most significant factor that contributes to the abandonment of assistive technology is that the device does not meet the users' needs or expectations.

King (1999) discussed the factors leading to the failure of assistive technologies, highlighting three major categories of failures. The first category relates to the people surrounding the user. For the people surrounding the user, correctly evaluating and communicating the users' condition is critical since many assistive technologies fail because the users' real needs are not known by the designers. The second category relates to the users themselves. From the perspective of the assistive technology users, design efforts should focus on matching the device's features to the user's needs, instead of the user to the device. The users' age, gender, and literacy can all play important roles in the adoption process. The third category relates to the assistive technology device itself. As to the assistive device, insufficient consideration of human factors during the design process has caused many assistive technologies to fail. Considering ease

of use and portability, the designers should focus on lighter and smaller solutions that provide easy access to the power supply. In addition, assistive devices need to be durable and repairable.

Kintsch and DePaula (2002) proposed a framework for facilitating the successful adoption of assistive technologies. The framework proposed four groups of participants in the adoption process: users, caregivers, designers, and assistive technology specialists. All four groups of participants need to be actively involved to support the adoption of assistive technologies. In the development phase, developers need to customize the product to the users' needs and abilities, make products simple and durable, and take into consideration user preferences. In the selection phase, products need to be assessed carefully by assistive technology specialists. The effectiveness of the product for the specific user must be evaluated, currently through a trial and reassessment process. In the learning phase, caregivers need to be trained with the product first so they can teach and support the user. In the integration phase, which is when the device becomes integrated into daily life, special patience is needed since tasks may initially require more time as the user gains experience.

Michaels and McDermott (2003) focused on the adoption of assistive technologies for students in the special education programs. The authors conducted interviews with employees from 356 special education programs and collected 143 surveys from special education teachers. The results show that the major barriers of assistive technology adoption in special education programs include funding, time limitations, lack of faculty knowledge or consistency in assistive technology focus, perceptions that assistive technology is only important for students with certain types of disabilities, and lack of short- and long-term plans.

Goette (2000) studied the use of speech-recognition technology in organizations and the key factors that contributed to successful adoption. A total of 40 individuals with PI or sensory disabilities who were using or had used speech recognition technologies were interviewed. Twenty-three of the participants had successfully adopted the speech-recognition technology, while 17 participants had abandoned it. The results suggest that participants who successfully adopted the technology had realistic expectations about the adoption process. Importantly, they expected to devote more time and effort than stated in the user manual to become proficient with the technology. For those users who had abandoned the technology, unrealistic employer expectations was one of the key reasons for abandonment.

CONCLUSIONS

The most obvious conclusion is that additional research is necessary. PI vary dramatically in terms of severity, temporal variability, and the body parts that are affected. Numerous technologies can be used in a variety of configurations. Given this variability and the limited number of studies reported in the literature, it is clear that only a fraction of the important questions have been investigated. Unfortunately, conducting informative studies involving individuals with PI can be difficult. Unlike computer users with no PI, the pool of potential participants is limited. Even when appropriate participants can be found, additional factors can make such studies difficult.

While additional research is needed, the existing literature does provide insights that can prove useful to both practitioners and researchers. Individually, the articles highlight potential benefits and limitations of various technologies for specific groups of individuals as well as new technologies that may prove useful for specific tasks or individuals in the future. When viewed as a whole, several lessons that extend beyond the technology or specific user groups studied become clear. We conclude by highlighting five such lessons.

PI Does Not Imply Disability

Impairments can, but do not always, result in disabilities (see Fig. 42.1). Several of the studies discussed earlier demonstrate this fact. Casali (1992) included three groups of participants including individuals with: PI resulting in low motor skills, PI resulting in high motor skills, and no PI. The high-motor-skills group was able to complete cursor manipulation tasks just as quickly as the group with no PI. This suggests that the high-motor-skills group did not experience any disability because of their PI. Similarly, Sears, Karat, Oseitutu, Karimullah, and Feng (2001) described a study in which individuals with high-level spinal-cord injuries were able to compose text just as quickly as individuals without PI. Several other studies provide examples where technology can either reduce or eliminate disabilities resulting from specific PI (e.g., Keates, Langdon, Clarkson, & Robinson, 2000). Overall, these results clearly indicate that it is possible, under the right circumstances, to design technology such that specific PI will not interfere with certain computing activities.

PI Affect Cognitive, Perceptual, and Motor Activities

PI are expected to affect an individual's ability to complete basic motor activities, but are not necessarily expected to affect performance on cognitive and perceptual activities. Many of the articles discussed earlier confirm differences in performance for motor activities (e.g., LoPresti, Brienza, Angelo, Gilbertson, & Sakai, 2000; Radwin, Vanderheiden, & Lin, 1990). More importantly, the unexpected affect of PI on cognitive and perceptual activities becomes apparent by analyzing both fundamental interactions and the high-level strategies adopted by individuals with PI.

Sears, Karat, Oseitutu, Karimullah, and Feng (2001) found that individuals with and without PI responded differently when using the same speech recognition system to accomplish a variety of tasks. Individuals with PI interrupted their dictation more often, spent a greater percentage of their time dictating as opposed to issuing commands, and navigated shorter distances. The results reported by Koester and Levine (1994) also highlight differences between individuals with and without PI with regard to both basic cognitive, perceptual, and motor activities as well as higher-level strategies. These studies confirm the importance of integrating individuals with PI into the process when designing new computer systems. Existing models of computer user behavior are based on data from individuals without PI, but it is clear that new models that integrate data from individuals with PI are necessary (Koester & Levine, 1994). These new models must address differences not only for motor activities, but for cognitive and perceptual activities as well.

Basic Actions Can Be Difficult

A number of articles have highlighted the difficulty individuals with PI may experience with even the most basic interaction activities. Trewin and Pain (1999) highlighted a variety of difficulties with activities as fundamental as pressing keys on a keyboard or pointing at and clicking on objects with a mouse. Casali (1992) also illustrated the difficulty individuals with PI could have with activities that require the mouse button to be held down while dragging the mouse. Unfortunately, these difficulties become even more significant as the target of the action gets smaller (Casali, 1992). For example, Radwin, Vanderheiden, and Lin (1990) showed that performance with head-controlled devices was affected by target size even more than the mouse.

Standardized Descriptions of Health Conditions and PI

Perhaps the greatest hindrance for both practitioners and researchers is the lack of accepted standards for describing the capabilities and limitations of the individuals who will make use of the new solutions that are developed. For studies including traditional computer users without PI, we typically provide basic demographics such as age, gender, and computer experience. For these users, we typically assume "normal" perceptual, cognitive, and motor abilities. Occasionally, researchers even may report that participants all had "normal" or "corrected to normal" vision or hearing if these abilities are particularly relevant for a given study. For individuals with PI (or cognitive or perceptual impairments for that matter), we need to describe the individual's perceptual, cognitive, and motor abilities. Unfortunately, existing studies rarely provide sufficient details regarding health conditions or associated PI. Many articles simply list a health condition for each participant (e.g., CP, choreoathetoid CP, Incomplete SCI at C6, Friedrich's Ataxia). While this provides a general sense as to the PI that may exist, it does not provide a sufficiently detailed understanding of the participants' abilities and limitations.

When a health condition results in a well-defined set of PI, a precise description of the underlying health condition may be sufficient. For example, a SCI at or above C5 with an ASIA score of A or B results in paralysis of the muscles used to control the arms and legs. Given this health condition, we know that an individual would not be able to use her hands or arms when interacting with a computer. In contrast, an SCI at or above C5, with an ASIA score of C or D, results in much more ambiguous PI. Similarly, indicating that an individual has CP provides general insights, but the resulting PI can vary dramatically. When a health condition does not map to a precise set of PI that are static, it is critical to describe both the health condition and the resulting PI (and any other impairments that may affect interactions with computers).

Health conditions are best described by physicians. Physical examinations can often provide the necessary information, but access to medical records may be sufficient. Ideally, a standardized scale will be used to describe the health condition, such as the ASIA scale that is used to describe the completeness of a spinal cord injury. However, such assessments focus on the status of the individual's health, not necessarily on their physical abilities. Understanding an individual's health can provide insights into changes that may occur in the future. Unfortunately, most of the time health conditions do not map directly to a well-defined set of PI.

In these situations, an occupational therapist can become involved in assessing each participant's PI. Occupational therapists can evaluate specific skills or abilities, including sensory motor, cognitive, and motor skills as necessary. The primary focus should be on those skills and abilities that may affect the individual's ability to interact with computers. While numerous standardized tests exist, many of these tests focus on activities that are not directly relevant to the use of information technologies. Assessments by occupational therapists focus on the impairments an individual experiences that may affect an individual's ability to perform certain activities without regard to the underlying health condition. These assessments focus on an individual's current status and therefore do not address changes that may occur due to the progression of a disease. New approaches for assessing and documenting the physical capabilities of individuals, that are relevant for interactions with information technologies, could prove useful in a variety of contexts. Effective solutions could be used by rehabilitation specialists as they identify appropriate technologies for patients as well as researchers seeking to provide more comprehensive descriptions of the participants in their research studies.

Successful Adoption of Assistive Technologies Is Critical

Assistive technologies are critical for helping users with PI interact with computer systems, but current adoption rates are highly disappointing. Scherer (1996) indicated that abandonment rates for assistive technologies range from 8% to 75%, with approximately one third of all assistive technologies being abandoned. It is also clear that the majority of new solutions are abandoned relatively soon after the initial interaction. King (1999) suggested that successful adoption of assistive technologies requires sufficient consideration of human factors when designing devices and active involvement of both the end-user and the people surrounding the user, such as the caregivers and healthcare providers. Kintsch and DePaula (2002) also emphasized the importance of active involvement by the end-users, caregivers, designers, and assistive technology specialists. To design and distribute effective assistive technologies that users will readily adopt, active user involvement and sufficient consideration for human factors is required.

FUTURE DIRECTIONS

The existing literature

- Highlighted the potential benefits and limitations of various technologies for specific groups of individuals;
- Provided examples of situations in which PI did not result in disabilities;
- Illustrated the potential impact of PI on cognitive, perceptual, and motor activities;

- Confirmed the difficulties individuals with PI may experience with basic actions such as pressing keys and clicking the mouse button, as well as more complex actions;
- Reenforced the need for standardized descriptions of health conditions and PI;
- Highlighted how infrequently assistive technologies are successfully adopted as well as some of the underlying problems that lead to abandonment.

This same literature confirmed the potential of various new technologies while highlighting the need for additional research. Several studies illustrated the use of body parts that are normally ignored when designing computer interfaces (e.g., Shaw, Loomis, & Crisman, 1995; Salem & Zhai, 1997). Others explored the potential of using electrophysiological data to control computers (e.g., Kennedy, Kakay, Moore, Adams, & Goldwaithe, 2000; Kübler, Kochoubey, Hinterberger, Ghanayim, Perelmouter, Schauer, et al., 1999; Doherty, Cockton, Bloor, & Benigno, 2000). These studies were exploratory, but they highlight the potential of these new technologies, especially for individuals with severe PI.

We must continue to explore technologies that support alternative methods of interacting with computers, but we also need a better understanding of both the potential and limitations of existing technologies. Existing studies provide examples where individuals with PI were just as fast and accurate as individuals with no PI, but important differences were also apparent for satisfaction ratings and the strategies that the individuals with PI employed. Other studies highlighted situations in which PI resulted in longer task completion times, higher error rates, or more subtle differences that only became apparent when performance was analyzed in detail. Future studies should attend to changes in cognitive, perceptual, and motor activities, which may reveal themselves through task completion times, error rates, satisfaction ratings, or through changes in the strategies that individuals adopt when completing a task. To allow both researchers and practitioners to more effectively interpret and generalize the results of such studies, we must carefully document both the health conditions and the PI of study participants. Ideally, new standardized methods for describing the physical capabilities of study participants will be developed and adopted.

While research into the efficacy of various technologies for individuals with PI is important, studies that assess the outcomes when assistive technologies are actually used are also critical (Fuhrer, 2001). Research studies highlight the potential of a technology, but do not guarantee the success of a technology when used in the field. Outcomes research will ultimately determine the success or failure of the technologies developed by the HCI community for individuals with PI. As a result, it is important for HCI researchers and system developers to work closely with those rehabilitation professionals who evaluate the needs of individuals with PI, determine which technologies they should be using, and assess the success or failure of these technologies.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants IIS-9910607, IIS-0121570, IIS-0328391, and IIS-0511954. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

References

ACM Conference on Universal Usability (CUU) (2003). Retrieved January 5, 2006, from http://sigchi.org/cuu2003/

ACM SIGACCESS Conference on Computers and Accessibility (ASSETS) (2006). Retrieved March 8, 2007, from http://www.acm.org/sigaccess/assets06/

Albacete, P. L., Chang, S. K., & Polese, G. (1994). Iconic language design for people with significant speech and multiple impairments. In *Proceedings of ASSETS 94* (pp. 23–30). New York: ACM.

Albacete, P. L., Chang, S. K., Polese G., & Baker, B. (1998). Iconic language design for people with significant speech and multiple impairments. In *Lecture notes in computer science: Vol. 1458. Assistive technology and artificial intelligence, applications in robotics, user interfaces and natural language processing* (pp. 12–32). Berlin Heidelberg: Springer-Verlag.

Allanson, J., Rodden, T., & Mariani, J. (1999). A toolkit for exploring electro-physiological human-computer interaction. In *Proceedings of INTERACT'99* (pp. 231–237). Amsterdam: IOS Press.

Alm, N., Todman, J., Elder, L., & Newell, A. F. (1993). Computer aided conversation for severely physically impaired non-speaking people. In *Proceedings of InterCHI 93* (pp. 236–241). New York: ACM.

American Spinal Injury Association (ASIA). (2001). *ASIA impairment scale*. Retrieved June 27, 2001, from http://www.asia-spinalinjury.org/publications/2001_Classif_worksheet.pdf

Bain, K., Bass, R., & Wald, E. (2002). Speech recognition in university classrooms: liberated learning project. In *Proceedings of ASSETS 2002* (pp. 192–199). Edinburgh: ACM.

Banerjee, S. (1982). *Rehabilitation management of amputees*. Baltimore: Williams & Wilkins.

Barreto, A. B., Scargle, S. D., & Adjouadi, M. (1999). A real-time assistive computer interface for users with motor disabilities. *ACM SIGCAPH Computers and the Physically Handicapped, 64*, 6–16.

Bates, R., & Istance, H. (2002). Zooming interfaces! Enhancing the performance of eye controlled pointing devices. In *Proceedings of ASSETS 2002* (pp. 119–126). Edinburgh.

Berkow, R. (1997). *The Merck manual of medical information home edition*. Whitehouse Station, NJ: Merck Research Laboratories.

Birbaumer, N., Hinterberger, T., Kübler, A., & Neumann, N. (2003, June). The thought-translation device (TTD): Neurobehavioral mechanisms and clinical outcome, *IEEE Trans. on Neural Systems and Rehabilitation Engineering, 11*(2), 120–123.

Birch, G., Bozorgzadeh, Z., & Mason, S. (2002). Initial on-line evaluations of the LF-ASD brain-computer interface with able-bodied and spinal-cord subjects using imagined voluntary motor potentials. *IEEE Trans Neural Systems and Rehabilitation Engineering, 10*(4), 219–24.

Boissiere, P., & Dours, D. (2000). VITIPI: A universal writing interface for all. *Proceedings of the 6th ERCIM Workshop: User Interfaces for All*.

Borisoff, J. F., Mason, S., Bashashati, A., & Birch, G. (2004). Brain-computer interface design for asynchronous control applications: Improvements to the LF-ASD asynchronous brain switch. *IEEE Transactions on Biomedical Engineering, 51*(6), 985–992.

Bowman, T. (1997). VR meets physical therapy. *Communications of the ACM, 40*(8), 59–60.

Bozorgzadeh, Z., Birch, G. E., & Mason, S. G. (2000). The LF-ASD BCI: On-line identification of imagined finger movements in spontaneous EEG with able-bodied subjects. *Proceedings of ICASSP 2000.* Vol. 4, pp. 2385–2388.

Card, S., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.

California State University, Northridge Center (CSUN) (2006). California State University, Northridge Center On Disabilities Annual Conference. Retrieved March 8, 2007, from http://www.csun.edu/cod/conf/2006/proceedings/csun06.htm

Carroll, K., Schlag, C., Kirikci, O., & Moore, M. (2002). Communication by neural control. In *Proceedings of CHI 2002* (pp. 590–591). Minneapolis, MN: ACM

Casali, S. P. (1992). Cursor control device used by persons with physical disabilities: Implications for hardware and software design. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 311–315). Santa Monica: HFES.

Casali, S. P. (1995). A physical skills based strategy for choosing an appropriate interface method. In A. D. N. Edwards (Ed.), *Extra-ordinary human-computer interaction* (pp. 315–341). Cambridge: Cambridge University Press.

Casali, S. P., & Chase, J. D. (1993). The effects of physical attributes of computer interface design on novice and experienced performance of users with physical disabilities. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting* (pp. 849–853). Santa Monica: HFES.

Closing the Gap: Computer technology in special education and rehabilitation. (2006). Retrieved March 8, 2007, from http://www.closingthegap.com/confTest/index.lasso

Craig, A., Tran, Y., McIsaac, P., Moses, P., Kirkup, L., & Searle, A. (2000). The effectiveness of activating electrical devices using alpha wave synchronization contingent with eye closure. *Applied Ergonomics, 31*, 377–382.

Dai, L., Goldman, R., Sears, A., & Lozier, J. (2004). Speech-based cursor control: A study of grid-based solutions. In *Proceedings of ASSETS 2004* (pp. 94–101). Atlanta: ACM.

De Silva, G., Lyons, M., Kawato, S., & Tetsutani, N. (2003). Human factors evaluation of a vision-based facial gesture interface. *Proceedings,of CVPR-HCI 2003*, Vol. 5. (pp. 52–59)..

Demasco, P. W., & McCoy, K. F. (1992). Generating text from compressed input: An intelligent interface for people with severe motor impairments. *Communications of the ACM, 35*(5), 68–78.

Demasco, P., Newell, A. F., & Arnott, J. L. (1994). The application of spatialization and spatial metaphor to augmentative and alternative communication. In *Proceedings of ASSETS 94* (pp. 31–38). New York: ACM.

Doherty, E., Cockton, G., Bloor, C., & Benigno, D. (2000). Mixing oil and water: Transcending method boundaries in assistive technology for traumatic brain injury. In *Proceedings of ASSETS 2000* (pp. 110–117). New York: ACM.

Doherty, E., Cockton, G., Bloor, C., & Benigno, D. (2001). Improving the performance of the Cyberlink mental interface with the 'Yes/No program'. *Proceedings of CHI 2001*, 69–76: ACM.

Duvaisin, R. (1991). *Parkinson's disease: A guide for patients and families*. New York: Raven Press.

Edwards, A. D. N. (1995). *Extra-ordinary human-computer interaction*. Cambridge: Cambridge University Press.

European Research Consortium for Informatics and Mathematics (ERCIM) (2006). 9th ERCIM Working Group: User Interface for All. Retrieved March 8, 2007, from http://ui4all.ics.forth.gr/workshop2006/

Evans, D. G., Drew, R., & Blenkhorn, P. (2000). Controlling mouse pointer position using an infrared head-operated joystick. *IEEE Transactions on Rehabilitation Engineering, 8*(1), 107–117.

Fell, H. J., Delta, H., Peterson, R., Ferrier, L. J., Mooraj, Z., & Valleau, M. (1994). Using the baby-babble-blanket for infants with motor problems: An empirical study. In *Proceedings of ASSETS 94* (pp. 77–84). New York: ACM.

Felzer, T., & Freisleben, B. (2002). HaWCoS: The hands-free wheel chair control system. In *Proceedings of ASSETS 2002* (pp. 127–134). Edinburgh: ACM.

Feng, J., & Sears, A. (2004). Using confidence scores to improve hands-free speech-based navigation in continuous dictation systems. *ACM Transactions on Computer-Human Interaction, 11*(4), 329–356.

Fuhrer, M. J. (2001). Assistive technology outcomes research: Challenges met and yet unmet. *American Journal of Physical Medicine and Rehabilitation, 80*(7), 528–535.

Garay-Vitoria, N., & González-Abascal, J. (1997). Intelligent word-prediction to enhance text input rate. In *Proceedings of IUI 97* (pp. 241– 244). New York: ACM.

Gnanayutham, P., Bloor, C., & Cockton, G. (2005). Discrete acceleration and personalized tiling as brain-body interface paradigms for neurorehabilitation. In *Proceedings of CHI 2005* (pp. 261–270). Portland: ACM.

Goette, T. (1998). Factors leading to the successful use of voice recognition technology. In *Proceedings of ASSETS 98* (pp. 189–196). New York: ACM.

Goette, T. (2000). Keys to the adoption and use of voice recognition technology in organizations. *Information Technology & People,* Vol. *13*(1), 67–80.

Gray, D. B., Quatrano, L. A., & Liberman, M. L. (1998). Designing and using assistive technology. Baltimore, MD: Paul H. Brooks Publishing Co.

Hallett, M. (1991). Classification and treatment of tremor. *Journal of the American Medical Association, 266*, 1115.

Hansen, J., Itoh, K., Tørning, K., Aoki, H., & Johanson, A. (2004). Gaze typing compared with input by head and hand. *Proceedings of the Eye Tracking Research & Applications Symposium on Eye Tracking Research and Applications,* 131–138.

Hewitt, J., Lyon, C., Britton, C., & Mellor, B. (2005). SpeakView: Live captioning of lectures. *Proceedings of HCII 2005*.

Hicks, J. E. (1988). Approach to diagnosis of rheumatoid disease. *Archives of Physical Medical Rehabilitation, 69*, S–79.

Horn, L. J., & Zasler, N. D. (Eds.). (1996). *Medical rehabilitation of traumatic brain injury.* Philadelphia: Hanley & Belfus.

Hornof, A., Cavender, A., & Hoselton, R. (2004). EyeDraw: A system for drawing pictures with eye movements. *Proceedings of ASSETS 2004,* 86–93. ACM.

Hwang, F., Keates, S., Langdon, P., & Clarkson J. (2003). Multiple haptic targets for motion-impaired computer users. In *Proceedings of CHI 2003* (pp. 41–48). Fort Lauderdale: ACM.

Hwang, F., Keates, S., Langdon, P., & Clarkson, J. (2004). Mouse movements of motion-impaired users: A submovement analysis. In *Proceedings of ASSETS 2004* (pp.102–109). Atlanta: ACM.

Karimullah, A., & Sears, A. (2002). Speech-based cursor control. *Proceedings of ASSETS 2002* (pp. 178–185). Edinburgh: ACM.

Keates, S., Clarkson, J., & Robinson, P. (2000). Investigating the application of user models for motion-impaired users. *Proceedings of ASSETS 2000* (pp. 129–136). New York: ACM.

Keates, S., Hwang, F., Langdon, P., & Clarkson, J. (2002). Cursor measures for motion-impaired computer users. In *Proceedings of ASSETS 2002* (pp. 135–142). Edinburgh: ACM.

Keates, S., Langdon, P., Clarkson, J., & Robinson, P. (2000). Investigating the use of force feedback for motor-impaired users. *Proceedings of the 6th ERCIM Workshop: User Interfaces for All*.

Keirn, Z. A., & Aunon, J. I. (1990). Man-machine communications through brain-wave processing. *IEEE Engineering in Medicine and Biology Magazine, 9*(1), 55–57.

Kennedy, P. R., Kakay, R. A. E., Moore, M. M., Adams, K., & Goldwaithe, J. (2000). Direct control of a computer from the human central nervous system. *IEEE Transactions on Rehabilitation Engineering, 8*(2), 198–202.

King, T. W. (1999). Why AT fails: A human factors perspective. In. T. King (Ed.), *Assistive technology, essential human factors* (pp. 231–255). Boston: Allyn and Bacon.

Kintsch, A., & DePaula, R. (2002). A framework for the adoption of assistive technology. *Proceedings of the SWAAAC 2002: Supporting Learning Through Assistive Technology, Winter Park, CO, USA,* pp. E3 1–10.

Kirkup, L., Searle, A., Craig, A., McIsaac, P., & Moses, P. (1997). EEG-based system for rapid on-off switching without prior learning. *Medical and Biological Engineering and Computing, 35*(5), 504–509.

Koester, H. (2004). Usage, performance, and satisfaction outcomes for experienced users of automatic speech recognition. *Journal of Rehabilitation Research and Development, 41*(5), 739–754.

Koester, H. H., & Levine, S. P. (1994). Validation of a keystroke-level model for a text entry system used by people with disabilities. In *Proceedings of ASSETS 94* (pp. 115–122). New York: ACM.

Kübler, A., Kochoubey, B., Hinterberger, T., Ghanayim, N., Perelmouter, J., Schauer, et al. (1999). The thought translation device: A neurophysiological approach to communication in total motor paralysis. *Experimental Brain Research, 124*(2), 223–232.

LaCourse, J. R., & Hludik, F. C. J. (1990). An eye movement communication-control system for the disabled. *IEEE Transactions on Biomedical Engineering, 37*(12), 1215–1220.

Lankford, C. (2000). Effective eye-gaze input into Windows. *Proceedings of Eye Tracking Research and Applications Symposium 2000,* pp. 23–27. ACM.

Lin, M. L., Radwin, R. G., & Vanderheiden, G. C. (1992). Gain effects on performance using a head-controlled computer input device. *Ergonomics, 35*(2), 159–175.

Lisogurski, D., & Birch, G. E. (1998). Identification of finger flexions from continuous EEG as a brain computer interface. *Proceedings of IEEE Engineering in Medicine and Biology Society 20th Annual International Conference.* Vol. 4. (pp. 2004–2007).

LoPresti, E. F. (2001). Effect of neck range of motion limitations on the use of head controls. In *CHI 2001 Extended Abstracts* (pp. 75–76). New York: ACM.

LoPresti, E., & Brienza, D. (2004) Adaptive software for head-operated computer controls. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12*(1), 102–111.

LoPresti, E., Brienza, D. M., Angelo, J., Gilbertson, L., & Sakai, J. (2000). Neck range of motion and use of computer head controls. In *Proceedings of ASSETS 2000* (pp. 121–128). New York: ACM.

Lusted, H. S., & Knapp, B. R. (1996). Controlling computers with neural signals. *Scientific American, 275*(4), 58–63.

Malkewitz, R. (1998). Head pointing and speech control as a hands-free interface to desktop computing. In *Proceedings of ASSETS 98* (pp. 182–188). New York: ACM.

Manaris, B., & Harkreader, A. (1998). SUITEKeys: A speech understanding interface for the motor-control challenged. In *Proceedings of ASSETS 98* (pp. 108–115). New York: ACM.

Manaris, B., Macgyvers, V., & Lagoudakis, M. (2002). A listening keyboard for users with motor impairments—a usability study. *International Journal of Speech Technology, 5*, 371–388.

Manasse, B., Hux, K., & Rankin-Erickson, J. (2000). Speech recognition training for enhancing written language generation by a traumatic brain injury survivor. *Brain Injury, 14*(11), 1015–1034.

Mason, S. G., Bozorgzadeh, Z., & Birch, G. E. (2000). The LG-ASD brain computer interface: On-line identification of imagined finger flexions in subjects with spinal cord injuries. In *Proceedings of ASSETS 2000* (pp. 109–113). New York: ACM.

Matias, E., MacKenzie, I. S., & Buxton, W. (1993). Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. In *Proceedings of InterCHI 93* (pp. 88–94). New York: ACM.

McCoy, K. F., Demasco, P. W., Jones, M. A., Pennington, C. A., Vanderheyden, P. B., & Zickus, W. M. (1994). A communication tool for people with disabilities: Lexical semantics for filling in the pieces. In *Proceedings of ASSETS 94* (pp. 107–114). New York: ACM.

McCoy, K., Demasco, P., Pennington, C., & Badman, A. (1997). Some interface issues in developing intelligent communication aids for people with disabilities. In *Proceedings of the International Conference on Intelligent User Interfaces* (pp. 163–170). Orlando: ACM.

Michaels, C., & McDermott, J. (2003). Assistive technology integration in special education teacher preparation: Program coordinators' perceptions of current attainment and importance. *Journal of Special Education Technology, 18*(3), 29–44.

Molnar, G. E. (1992). Cerebral palsy. In G. E. Molnar (Ed.), *Pediatric rehabilitation* (pp. 481–533). Baltimore: Williams & Wilkins.

Moore, M., & Dua, U. (2004). A galvanic skin response interface for people with severe motor disabilities. In *Proceedings of ASSETS 2004* (pp. 48–54). Atlanta: ACM.

Moore, J. S., & Garg, A. (Eds.). (1992). Ergonomics: Low-back pain, carpal tunnel syndrome, and upper extremity disorders in the workplace. *State of the Art Reviews Occupational Medicine, 7*(4), 593–790.

Moore, M., & Kennedy, P. (2000). Human factors issues in the neural signals direct brain-computer interface. In *Proceedings of ASSETS 2000* (pp. 114–120). Arlington: ACM.

Myers, B., Wobbrock, J., Yang, S., Yeung, B., Nichols, J., & Miller, R. (2002). Using handhelds to help people with motor impairments. In *Proceedings of ASSETS 2002* (pp. 89–96). Edinburgh: ACM.

Oakley, I., McGee, M. R., Brewster, S., & Gray, P. (2000). Putting the feel in 'Look and Feel'. In *Proceedings of CHI 2000* (415–422). New York: ACM.

Oseitutu, K., Feng, J., Sears, A., & Karat, C.-M. (2001). Speech recognition for data entry by individuals with spinal cord injuries. *Proceedings of the 1st International Conference on Universal Access in Human-Computer Interaction*. Vol. 3. (pp. 402–406)

Patmore, D. W., & Knapp, R. B. (1995). A cursor controller using evoked potentials and EOG. In *Proceedings of RESNA 95 Annual Conference* (pp. 702–704). Arlington, VA: RESNA.

Patmore, D. W., & Knapp, R. B. (1998). Towards an EOG-based eye tracker for computer control. In *Proceedings of ASSETS 98* (pp. 197–203). New York: ACM.

Pidcock, F. S., & Christensen, J. R. (1997). General and neuromuscular rehabilitation in children. In B. O'Young, M. Young, & S. Stiens (Eds.), *PM&R Secrets* (pp. 402–406). Philadelphia: Hanley & Belfus.

Price, K. J., Lin, M., Feng, J., Goldman, R., Sears, A., & Jacko, J. A. (2006). Motion does matter: An examination of speech-based text entry on the move. *Universal access in the information society*. Vol 4 (3), March 2006. 246–257.

Pieper, M., & Kobsa, A. (1999). Talking to the ceiling: An interface for bed-ridden manually impaired users. In *CHI 1999, Extended Abstract* (pp. 9–10). Pittsburg: ACM.

Radwin, R. G., Vanderheiden, G. C., & Lin, M. L. (1990). A method for evaluating head-controlled computer input devices using Fitts' Law. *Human Factors, 32*(4), 423–438.

Rehabilitation Engineering and assistive technology Society of North America (RESNA) (2006). Retrieved March 8, 2007, from http://www.resna.org/

Ropper, A. & Brown, R. (2005). Adams and Victor's Principles of Neurology, 8th Edition. The McGraw-Hill Companes, Inc.

Rosenthal, M., Griffith, E. R., Bond, M. R., & Miller, J. D. (Eds.). (1990). *Rehabilitation of the adult and child with traumatic brain injury* (2nd ed.). Philadelphia: F. A. Davis.

Roy, D. M., Panayi, M., Erenshteyn, R., Foulds, R., & Fawcus, R. (1994). Gestural human-machine interaction for people with severe speech and motor impairment due to cerebral palsy. In *CHI 94 Conference Companion* (pp. 313–314). New York: ACM.

Salem, C., & Zhai, S. (1997). An isometric tongue pointing device. In *Proceedings of CHI 97* (pp. 538–539). New York: ACM.

Schaab, J. A., Radwin, R. G., Vanderheiden, G. C., & Hansen, P. K. (1996). A comparison of two control-display gain measures for head-controlled computer input devices. *Human Factors, 38*(3), 390–403.

Scherer, M. J. (1996). Dilemmas, challenges, and opportunities. In M. Scherer (Ed.), *Living in the state of struck* (pp. 115–121). Cambridge, Massachusetts: Brookline Books.

Scherer, M. J., & Galvin, J. C. (1996). An outcome perspective of quality pathways to most appropriate technology. In J. C. Galvin & M. J. Scherer (Eds.), *Evaluating, selecting and using appropriate assistive technology* (pp. 1–26). Gaithersburg, MD: Aspen Publication.

Schumacher, H. R. Jr. (Ed.). (1993). *Primer on the rheumatic diseases*. Atlanta: Arthritis Foundation.

Sears, A., Feng, J., Oseitutu, K., & Karat, C. M. (2003). Speech-based navigation during dictation: Difficulties, consequences, and solutions. *Human Computer Interaction, 18*(3), 229–257.

Sears, A., Karat, C.-M., Oseitutu, K., Karimullah, A., & Feng, J. (2001). Productivity, satisfaction, and interaction strategies of individual with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society, 1*, 1–12.

Sears, A., Lin, M., & Karimullah, A. (2002) Speech-based cursor control: understanding the effects of target size, cursor speed, and command selection. *Universal Access in Information Society*, Vol. 2(1), 30–43.

Shaw, R., Loomis, A., & Crisman, E. (1995). Input and integration: Enabling technologies for disabled users. In A. D. N. Edwards (Ed.), *Extra-ordinary human-computer interaction* (pp. 263–277). Cambridge: Cambridge University Press.

Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. In *Proceedings of CHI 2000* (pp. 281–288). New York: ACM.

Stephanidis, C. (2001). *User interfaces for all: Concepts, methods, and tools*. Mahwah, NJ: Lawrence Erlbaum Associates.

Stiens, S., Goldstein, B., Hammond, M., & Little, J. (1997). Spinal cord injuries. In B. O'Young, M. Young, & S. Stiens (Eds.), *PM&R Secrets* (pp. 253–261). Philadelphia: Hanley & Belfus.

Taylor, R. G., & Lieberman, J. S. (1988). Rehabilitation of patient with diseases affecting the motor unit. In J. A. DeLisa (Ed.), *Rehabilitation medicine: Principles and practice* (pp. 811–820). Philadelphia: J. B. Lippincott.

Trace Center (2007). Trace Center, College of Engineering, University of Wisconsin—Madison. Retrieved March 8, 2007, from http://www.trace.wisc.edu

Trewin, S. (1996). A study of input device manipulation difficulties. In *Proceedings of ASSETS 96* (pp. 15–22). New York: ACM.

Trewin, S. (2002) An invisible keyboard. In *Proceedings of ASSETS 2002* (pp. 143–149). Edinburgh: ACM.

Trewin, S. (2004). Automating accessibility: The dynamic keyboard. In *Proceedings of ASSETS 2004* (pp. 71–78). Atlanta: ACM.

Trewin, S., & Pain, H. (1998a). A study of two keyboard aids to accessibility. In *Proceedings of the HCI'98 Conference on People and Computers XIII* (83–97). Heidelberg: Springer-Verlag.

Trewin, S., & Pain, H. (1998b). A model of keyboard configuration requirements. In *Proceedings of ASSETS 98* (pp. 173–181). New York: ACM.

Trewin, S., & Pain, H. (1999). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies, 50*, 109–144.

Trewin, S., Slobogin, P., & Power, M. (2002). Accelerating assessment with self-optimizing devices. *Proceedings of ICCHP 2002,* 273– 275.

Wade, D. T., Langton Hewer, R., Skilbeck, C. E., & David, R. M. (1985). *Stroke: A critical approach to diagnosis, treatment, and management*. Chicago: Year Book.

Wade, T., Langton Hewer, R., Skilbeck, C., & David, R. (1985). *Stroke: A critical approach to diagnosis, treatment, and management*. London: Chapman and Hall.

World Health Organization. (2006). *International classification of functioning, disability, and health*. Retrieved February 20, 2006, from http://www3.who.int/icf/

Willis, T., Pain, H., Trewin, S., & Clark, S. (2002). Informing flexible abbreviation expansion for users with motor disabilities. *Proceedings of International Conference on Computers Helping People with Special Needs* (*ICCHP*) *2002,* 251–258.

Wobbrock, J., Aung, H., Myers, B., & LoPresti, E. (2004). Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads. In *Proceedings of ASSETS 2004* (pp. 110–117). Atlanta: ACM.

Young, M. A., Tumanon, R. C., & Sokal, J. O. (2000, Summer). Independence for people with disabilities: A physician's primer on assistive technology. *Maryland Medicine,* 28–32.

·43·

PERCEPTUAL IMPAIRMENTS: NEW ADVANCEMENTS PROMOTING TECHNOLOGICAL ACCESS

Julie A. Jacko
Georgia Institute of Technology

V. Kathlene Leonard
Alucid Solution, Inc.

Ingrid U. Scott
Penn State College of Medicine

Introduction **854**
Visual Function **855**
Definitions 855
Epidemiology 855
Selected Visual Disorders 856
Age-related macular degeneration 856
Diabetic retinopathy 856
Glaucoma 857
Specific Visual Functions 857
Visual acuity 857
Contrast sensitivity 857
Visual field 857
Color vision 858
Visual Function and Age 858
Summary 858
Highlighting Technological Advancements in HCI Research **859**
Perceptual Interfaces 859
Multimedia interfaces 861
Multimodal interfaces 861
Adaptive Interfaces 863
Summary 866
Conclusions **866**
Acknowledgments **866**
References **866**

INTRODUCTION

The introduction of early computers facilitated new ways for individuals with visual impairment to access information electronically: magnified, in Braille, or aurally via the conversion of digital information. However, the introduction of GUIs that present digital information via visual metaphors and icons contributed to the digital divide, which can hamper the productivity of this population. In many cases, even access to documents and forms can be a difficult to impossible task when available through the direct manipulation paradigm (Fortuin & Omata, 2004). The exclusive reliance of GUIs on the visual interaction paradigm therefore threatens to limit accessibility for anyone whose visual channel is compromised (Dix, Finlay, Abowd, & Beale, 1998). This chapter provides readers with (a) an introduction to the visual sensory channel; (b) a review of research approaches, models, and theories that are relevant to HCI and visual impairment; and (c) a discussion of forms of visual dysfunction in the research, design, and evaluation of human-computer systems.

The interaction strategies and related interaction barriers for individuals with visual impairments in the past 15 years has received growing attention in an attempt to inform judicious, inclusive design for accessible information technologies (e.g., Assistive Technologies for Independent Aging: Opportunities and Challenges, 2004; Arditi, 2002; Brewster, Wright, & Edwards, A. D. N., 1994; Craven, 2003; Fortuin & Omata, 2004; Fraser & Gutwin, 2000; Gaver, 1989; Jacko, 1999; Jacko, Barnard, et al., 2004; Jacko, Barreto, Scott, Chu, et al., 2002; Jacko, Barreto, Scott, Rosa, & Pappas, 2000; Jacko, Moloney, et al., 2005; Jacko, Rosa, Scott, Pappas, & Dixon, 1999; Jacko & Sears, 1998). Visual impairments (that do not lead to blindness) can create barriers to distinguishing fine details of iconic screen targets and to tracking the highly dynamic nature of the pointer used to manipulate these icons (Fraser & Gutwin, 2000). This is largely attributed to difficulty in manipulating objects with the pointer due to reduced visual acuity and visual field.

In terms of the visual sensory channel, it is known that user behavior is strongly influenced by the nature and amount of residual vision the user experiences in combination with computer interface characteristics. As an extreme example, a blind user without any functional vision will use fundamentally different coping skills to navigate an interface as compared to an individual with clouded vision due to cataracts (Jacko & Sears, 1998). Harper, Goble, and Stevens (2001) emphasized that the differences in orientation, navigation, travel, and mobility of visually impaired versus sighted individuals should be considered in the design of technology because there are differences in the mental map and cognitive processes that occur across the spectrum of visual abilities.

The impetus for this chapter is two parts. First, the number of individuals who report low vision is anticipated to rise sharply with the aging baby boomers (who are, on average, living longer) as they experience age-related changes to their functional vision (e.g., reduced visual acuity, presbyopia, contrast sensitivity, color sensitivity, depth perception, and glare sensitivity). They are increasingly predisposed to acquire ocular diseases associated with older age (e.g., macular degeneration, diabetic retinopathy, glaucoma, and cataracts; for a review, see Orr, 1998; Schieber, 1994). Secondly, the digital divide imposed on the population with visual impairments has been measured in terms of technology access and unemployment (Gerber & Kirchner, 2001). Looking beyond the United States, as the need for information increases globally so does the diversity of the people requiring access. As a result, a potentially large number of users may be disadvantaged with respect to gaining access to a variety of types of information without adequate accommodations.

The framework for the structure of this chapter results from an HCI approach first introduced by Jacko and Vitense (2001), and further clarified by Jacko, Vitense, and Scott (2002). Initial work in this research area included a comprehensive review of the literature to facilitate the development of a categorization scheme to account for categories of impairment. From this literature review, five major categories emerged: (a) hearing impairments, (b) mental impairments, (c) physical impairments, (d) speech impairments, and (e) visual impairments. Fig. 43.1 illustrates that each of the five, overarching categories is composed of a collection of clinical diagnoses unique to that category (depicted in Fig. 43.1 by A_1,...,A_n, B_1,...,B_n, C_1,...,C_n, D_1,..., D_n and E_1,...,E_n). Each diagnosis, in turn, influences certain functional capabilities that are critical to the access of information technologies (depicted in Fig. 43.1 by Y_1,...,Y_n). A subset of these functional capabilities can be directly linked to specific classes of technologies (shown at the bottom of the diagram).

While this framework is applicable to the five identified categories of impairment, its discussion and demonstrated utility in the scope of this chapter will address only visual impairment. Consider, for example, a person who has been diagnosed with a specific type of visual impairment represented as E_1 in Fig. 43.1. This visual impairment results in measurable decrements to certain aspects of this person's functional visual capabilities, Y_7 and Y_8. Observe from Fig. 43.1 that the decrement to the functional capability represented Y_8 does not impede a person's access to any of the technology classes depicted in Fig. 43.1. In contrast, the functional visual capability represented by Y_7 in Fig. 43.1, which impedes this person's ability to successfully

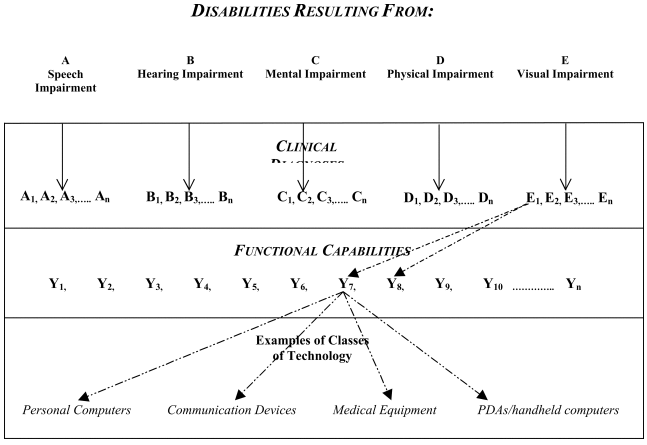


FIGURE 43.1. Framework for the Integration of Clinical Diagnoses, Functional Capabilities, and Access to Classes of Technologies (Adapted from Jacko, Vitense, & Scott, 2002).

access information using all four examples of information technology. From this conceptual representation, it is apparent that much more knowledge is needed for researchers to possess an accurate depiction of the empirical relationships that exist between diagnoses, functional capabilities, and access to specific classes of technologies. More specifically, emerging from this conceptual framework are several key research areas in need of investigation:

1. It is critical to establish empirical links between clinical diagnoses and sets of functional capabilities.
2. It is necessary to define the set of functional capabilities required to access information technologies.
3. It is essential to establish empirical bases for the influence of specific functional capabilities on access to specific classes of technologies.

While the framework illustrates five categories of impairment, this chapter will focus on visual impairment and functional visual capability. Auditory/speech, cognitive impairments and physical (motor) impairments are covered in this handbook in accompanying chapters.

Following this framework, this chapter introduces readers to the clinical definition of visual impairments and diagnoses, in which visual function is first addressed, followed by a discussion of the leading causes of vision loss in the United States and beyond. The section also provides a discussion of specific visual functions. Then, the chapter highlights recent advancements in HCI research, which effectively link the three major areas of the framework: clinical diagnosis, functional impairment, and interaction across a variety of classes of technology and impairments. This research is done in pursuit of HCI solutions for the visually impaired including perceptual interfaces, multimedia interfaces, multimodal interfaces, and adaptive interfaces. Finally, examples of technological advancements for this population are presented. While these technological advancements are discussed briefly in this chapter within the context of visual impairments, it should be noted that additional information on HCI and perceptual-motor interactions, hearing and speech impairments, and motor impairments can be found in their respective chapters.

VISUAL FUNCTION

Definitions

When considering the impact of visual loss on an individual's ability to use a computer effectively, it is first necessary to understand various dimensions of visual performance. Many different terms have been used to refer to abnormalities in visual function, including *disorder, impairment, disability*, and *handicap* (Colenbrander, 1977). Although often used as synonyms, there are distinct differences. For instance, while *disorder* and *impairment* describe aspects of an organ's condition, the term *disability* describes aspects of a patient's condition.

Disorder refers to an abnormality in the anatomy or physiology of an organ and, in the case of a visual disorder, may occur anywhere in the visual system. Examples of visual disorders include corneal scar, cataract, macular degeneration, optic atrophy, or occipital stroke. It is important to recognize that knowing the specific visual disorder provides no information concerning the functional capacity of the eye.

Impairment refers to a functional abnormality in the organ. Thus, varying degrees of visual impairment can be measured in terms of specific visual functions, such as visual acuity, contrast sensitivity, visual field, or color vision. While such impairment measures demonstrate how well the eye functions, they do not reveal the impact of the visual disorder on the patient's ability to perform everyday activities. For example, a physician may state that the patient's visual acuity has dropped by four lines on the eye chart, while the patient reports an inability to see well enough to use a computer.

Disability refers to the ability of a patient (rather than an organ) to perform tasks, such as daily living skills, vocational skills, reading, writing, mobility skills, and so on. Since disability implies a broader perspective (the focus is on the person as a whole rather than on a specific organ), it is no longer entirely vision specific. For instance, while computer skills may be reduced due to vision loss, they may also suffer due to such conditions as arthritis. It is the combination of visual and nonvisual skills that determines the abilities or disabilities of an individual. Vision substitution techniques (such as the use of a white cane and increased reliance on memory and on hearing) may be helpful in improving the ability of an individual to perform specific tasks.

A disorder may cause impairments, and impairments may cause disability. However, these links are not rigid. An analysis of these various dimensions of vision loss permits identification of interventions at each link, which may improve the functional status and quality of life of an individual with visual loss (Fig. 43.1; Fletcher, 1999). For example, if one were interested in improving the ability of an individual with vision loss to use a computer effectively, possible interventions include medical and surgical intervention to impact the visual disorder and impairment, visual aids and adaptive devices to impact visual impairment, and social interventions, training, counseling, and education to impact visual disability. The design of a computer interface that enhances the ability of an individual with vision loss to perceive graphical and textual information would have a beneficial effect on the degree of impairment and resulting level of disability of that individual.

Epidemiology

Low vision has been defined as a permanent visual impairment that is not correctable with glasses, contact lenses, or surgical intervention, and which interferes with normal everyday functioning (Mehr & Freid, 1975). Specifically, low vision is defined as a best-corrected visual acuity worse than 20/40 in the better-seeing eye (The Eye Diseases Prevalence Research Group, 2004a). *Blindness* is defined as a best-corrected visual acuity of 20/200 or worse according to the United States definition and is defined as a best-corrected visual acuity of worse than 20/400 according to the World Health Organization definition (The Eye Diseases Prevalence Research Group, 2004a). Based on demo-

graphics from the 2000 U.S. Census, an estimated 937,000 (0.78%) Americans older than 40 years were blind (United States definition); an additional 2.4 million Americans (1.98%) has low vision (The Eye Diseases Prevalence Research Group, 2004a). Thus, blindness or low vision affects approximately 1 in 28 Americans older than 40 years (The Eye Diseases Prevalence Research Group, 2004a). Largely due to the aging of the U.S. population, it is estimated that the number of blind persons older than 40 years in the United States will increase by approximately 70% to 1.6 million (prevalence of 1.1%) by the year 2020 (The Eye Diseases Prevalence Research Group, 2004a). Vision loss has been ranked third, behind arthritis and heart disease, among conditions that cause persons older than 70 years to need assistance in activities of daily living (LaPlante, 1988).

Selected Visual Disorders

In the United States, the most common causes of decreased vision are age-related macular degeneration (AMD), diabetic retinopathy, and glaucoma.

Age-related macular degeneration. AMD is the leading cause of irreversible visual loss in the western world in individuals over 60 years of age. The macula is the part of the retina that is responsible for central vision. AMD affects more than 1.75 million individuals in the United States, and due to the rapid aging of the U.S. population, this number is estimated to increase to almost 3 million by the year 2020 (The Eye Diseases Prevalence Research Group, 2004b). The prevalence of severe visual loss due to AMD increases with age. In the United States, at least 10% of persons between the ages of 65 and 75 years have lost some central vision due to AMD; among individuals over the age of 75 years, 30% have vision loss due to AMD.

Risk factors for this disease and its progression include age, sunlight exposure, smoking, ocular and skin pigmentation, elevated blood pressure, and elevated serum cholesterol levels. The role of nutrition has not been fully identified as a risk factor, but a diet low in antioxidants and lutein may be a contributing factor.

AMD is a bilateral disease in which visual loss in the first eye usually occurs at about 65 years of age; the second eye becomes involved at the rate of approximately 10% per year. The two main types of AMD are atrophic and exudative. The atrophic (dry) form of the disease is generally a slowly progressive disease that accounts for approximately 90% of cases. It is characterized by the deposition of abnormal material beneath the retina (drusen) and by degeneration and atrophy of the central retina (also known as the macula); patients typically note slowly progressive central visual loss. Although much less common, the exudative (wet) form of the disease is responsible for about 88% of legal blindness attributed to AMD. This form of the disease, which often occurs in association with atrophic AMD, is characterized by the growth of abnormal blood vessels beneath the central retina (macula); these abnormal blood vessels elevate and distort the retina and may leak fluid and blood beneath or into the retina. Vision loss may be sudden onset and rapidly progressive (in contrast to the atrophic form of the disease, where vision loss generally occurs progressively over several months or years). AMD can cause profound loss of central vision, but the disease generally does not affect peripheral vision, and therefore, patients typically retain their abilities to ambulate independently.

Currently, no proven treatment reverses the retinal damage that has already occurred due to AMD. In order to try to prevent further vision loss, recommendations made to patients include eye protection against ultraviolet light exposure (sunglasses with ultraviolet light protection), no smoking, optimal control of blood pressure and serum cholesterol level, and a diet rich in dark green leafy and orange vegetables (antioxidants are believed to reduce the damaging effects of light on the retina through their reducing and free-radical scavenging actions; lutein is a macular pigment). The Age-Related Eye Disease Study (AREDS) was a randomized, controlled clinical trial that demonstrated a statistically significant reduction in rates of at least moderate visual acuity loss in persons with moderate AMD who received supplementation with vitamins C and E, beta carotene, and zinc compared to persons with moderate AMD assigned to placebo (Age-Related Eye Disease Study Research Group, 2001). Laser photocoagulation and photodynamic treatment of the abnormal blood vessels found in patients with exudative macular degeneration may help to prevent severe vision loss in some cases. Intravitreal injections of pegaptanib, a pegylated modified oligonucleotide that binds to extracellular vascular endothelial growth factor (VEGF) isoform 165 (the isoform widely considered to be the primary pathologic form of VEGF), are associated with a higher likelihood of visual preservation and a slowing of visual loss among patients with exudative AMD (Gragoudas, Adamis, Cunningham, Feinsod, & Guyer, 2004). Surgical rotation of the retina away from the area of abnormal blood vessels has also been effective in some cases. Other treatment modalities currently under investigation include such drugs as corticosteroids, other anti-VEGF agents, and combination treatments.

Diabetic retinopathy. Approximately 16 million Americans suffer from diabetes mellitus, most of whom will develop diabetic retinopathy within 20 years of their diagnoses. In fact, after 20 years of diabetes, nearly 99% of those with insulin-dependent diabetes mellitus and 60% with non-insulin-dependent diabetes mellitus have some degree of diabetic retinopathy. Diabetic retinopathy is the leading cause of legal blindness in Americans aged 20 to 65 years, with 10,000 new cases of blindness annually. One million Americans have proliferative diabetic retinopathy, and 500,000 have macular edema. Among an estimated 10.2 million American adults aged 40 years and older known to have diabetes mellitus, the estimated crude prevalence rates for retinopathy and vision-threatening retinopathy are 40.3% and 8.2%, respectively (The Eye Diseases Prevalence Research Group, 2004b). The estimated U.S. general population prevalence rates for retinopathy and vision-threatening retinopathy are 3.4% (4.1 million persons) and 0.75% (899,000 persons), respectively (The Eye Diseases Prevalence Research Group, 2004c).

The major risk factor for diabetic retinopathy is duration of diabetes; it is estimated that at 15 years, 80% of diabetics will have background retinopathy and that, of these, 5% to 10% will progress to proliferative changes. Other risk factors include

long-term diabetic control (as reflected in serum levels of glycosylated hemoglobin), hypertension, smoking, and elevated serum cholesterol.

There are two main types of diabetic retinopathy: nonproliferative and proliferative. Nonproliferative diabetic retinopathy refers to retinal microvascular changes that are limited to the confines of the retina and include such findings as microaneurysms, dot and blot intraretinal hemorrhages, retinal edema, hard exudates, dilation and bleeding of retinal veins, intraretinal microvascular abnormalities, nerve fiber layer infarcts, arteriolar abnormalities, and focal areas of capillary nonperfusion. Nonproliferative diabetic retinopathy can affect visual function through two mechanisms: intraretinal capillary closure resulting in macular ischemia, and increased retinal vascular permeability resulting in macular edema. Clinically significant macular edema is defined as any one of the following: (a) retinal edema located at or within 500 um of the center of the macula; (b) hard exudates at or within 500 um of the center if associated with thickening of adjacent retina; and (c) a zone of retinal thickening larger than one optic disc area if located within one disc diameter of the center of the macula.

Proliferative diabetic retinopathy is characterized by extraretinal fibrovascular proliferation; that is, fibrovascular changes that extend beyond the confines of the retina and into the vitreous cavity. Fibrovascular proliferation in proliferative diabetic retinopathy may lead to tractional retinal detachment and vitreous hemorrhage. High-risk proliferative diabetic retinopathy is defined by any combination of three of the four following retinopathy risk factors: (a) presence of vitreous or preretinal hemorrhage; (b) presence of new vessels; (c) location of new vessels on or near the optic disc; and (d) moderate to severe extent of new vessels (Diabetic Retinopathy Study Research Group, 1979).

Management of diabetic retinopathy includes referring the patient to an internist for optimal glucose and blood pressure control. In the Early Treatment Diabetic Retinopathy Study, focal or grid laser photocoagulation treatment for clinically significant macular edema reduced the risk of moderate visual loss, increased the chance of visual improvement, and was associated with only mild loss of visual field (Early Treatment Diabetic Retinopathy Study Research Group, 1995). Intravitreal triamcinolone acetonide and other intravitreal anti-VEGF agents are currently under investigation for the treatment of diabetic macular edema. Panretinal laser photocoagulation treatment of eyes with high-risk proliferative diabetic retinopathy reduced the risk of severe visual loss by 50% compared to untreated control eyes (Diabetic Retinopathy Study Research Group, 1981). Surgery is often indicated for nonclearing vitreous hemorrhage and for tractional retinal detachment involving or threatening the macula.

Glaucoma. Primary open angle glaucoma (POAG) is the most prevalent type of glaucoma, affecting 1.3% to 2.1% of the general population over the age of 40 years in the United States. In the United States, the disease is the leading cause of irreversible blindness among Blacks and the third leading cause among Whites (following AMD and diabetic retinopathy), and is responsible for 12% of legal blindness. Risk factors for the disease include increasing age (especially greater than 40 years), African ethnicity, positive family history of glaucoma, diabetes mellitus, and myopia (nearsightedness).

POAG is a chronic, slowly progressive optic neuropathy characterized by atrophy of the optic nerve and loss of peripheral vision. Central vision is typically not affected until late in the disease. Because central vision is relatively unaffected until late in the disease, visual loss generally progresses without symptoms and may remain undiagnosed for quite some time. While usually bilateral, the disease may be quite asymmetrical. POAG is associated with increased intraocular pressure, but normal-tension glaucoma may cause glaucomatous vision loss in patients with normal intraocular pressure. Thus, normal eye pressure does not rule out the presence of glaucoma.

Treatment of POAG includes topical or systemic medications, laser, or surgery to lower the intraocular pressure to a level at which optic nerve damage no longer occurs. Visual field testing is performed regularly in order to evaluate for progressive loss of peripheral vision, and the optic nerve is examined regularly to evaluate for evidence of progressive optic atrophy (clinical signs of glaucoma in the optic disc include asymmetry of the neuroretinal rim, focal thinning of the neuroretinal rim, optic disc hemorrhage, and any acquired change in the disc rim appearance or the surrounding retinal nerve fiber layer).

Specific Visual Functions

Visual acuity. Visual acuity is the most common measure of central visual function and refers to the smallest object resolvable by the eye at a given distance. It is defined as the reciprocal of the smallest object size that can be recognized. Visual acuity is expressed as a fraction in which the numerator is the distance at which the patient recognizes the object and the denominator is the distance at which a standard eye recognizes the object. For instance, a visual acuity of 20/60 means that the patient needs an object three times larger or three times closer than a standard eye requires. The traditional visual acuity chart presents symbols of decreasing size with fixed high contrast. The visual acuity chart used most often in the clinical setting is the Snellen acuity chart, which is comprised of certain letters of the alphabet; the size of the letters are constant on a given line of the eye chart, and decrease in size the lower the line on the chart. In accurate Snellen notation, the numerator indicates the test distance and the denominator indicates the letter size seen by the patient.

Contrast sensitivity. Contrast sensitivity refers to the ability of the patient to detect differences in contrast and is defined as the reciprocal of the lowest contrast that can be detected. This may be measured with the Pelli-Robson chart, in which letters decrease in contrast rather than size, or the Bailey-Lovie chart, in which letters of a fixed low contrast are varied in size. Contrast sensitivity is considered a more sensitive indicator of visual function than Snellen acuity and may provide earlier detection of such pathology as retinal and optic nerve disease.

Visual field. Visual field is classically defined as a three-dimensional graphic representation of differential light sensitivity at different positions in space. *Perimetry* refers to the clinical

assessment of the visual field. Typically, visual field is assessed with kinetic or static perimetry. During kinetic perimetry, a test object of fixed intensity is moved along several meridians toward fixation and points where the object is first perceived are plotted in a circle. During static perimetry, a stationary test object is increased in intensity from below threshold until perceived by the patient, and threshold values yield a graphic profile section. While peripheral visual field loss often produces difficulty for patients in orientation and mobility functions, macular field loss (either centrally or paracentrally) often causes difficulties with reading. For instance, the presence of central or paracentral visual field loss is a more powerful predictor of reading speed than is visual acuity (Fletcher, Schuchard, Livingstone, Crane, & Hu, 1994).

Color vision. Evaluation of color vision may be performed using pseudoisochromatic color plates, which are quick and commonly available; they consist of circles in various colors such that a person with normal color vision function can distinguish a number from the background pattern of circles. Ishihara or Hardy-Rand-Rittler pseudoisochromatic color plates are designed to screen for congenital red/green color deficiencies, while Lanthony tritan plates may be used to detect blue/yellow defects, which are frequently present in acquired disease. With the Farnsworth-Munsell 100-hue test, the patient must order 84 colored disks; the time-consuming nature of this test limits its clinical use. The Farnsworth Panel D-15 is a shorter and more practical version (using 15 disks), but is less sensitive. Most color-vision defects are nonspecific.

Visual Function and Age

This section provides an overview of how levels of visual function vary with age, and to what degree. Aging is synonymous with natural declines in a person's sensory abilities. As such, the process of aging is accompanied by changes to the eye, including the retina and visual nervous system that can impact functional vision (Schieber, 1994). Additionally, older adults are more likely to acquire ocular conditions that can compromise visual functioning beyond normally anticipated changes, such as macular degeneration, diabetic retinopathy, and cataracts. Age-related vision loss commonly impinges on the ability to complete near vision tasks such as reading and using the computer (Arditi, 2004). An understanding of these functional declines provides direction for strategies aimed to mitigate the negative impact of these changes. HCI designers, developers, and usability specialists should be fully aware of these needs, as the needs of this growing user population will become an increased priority with the shift in demographics of population segment.

Aspects of visual function that are known to normally decline as part of the aging process include,

- Visual acuity
- Visual field
- Contrast sensitivity
- Color perception
- Floaters
- Dry eyes
- Increased need for light
- Difficulty with glare
- Dark/light adaptation
- Reduced depth perception (Orr, 1998)

Beyond these factors, eye movement efficiency and accuracy are observed to decline with old age. Older adults are known to be less accurate and/or slower in locating a target in the peripheral vision (see also Kline, D. W., & Scialfa, 1997; Lee, Legge, & Oritz, 2003). Age-related differences have also been observed with the effectiveness with which older adults visually track targets with higher velocities. In both cases, these trends are typically aggravated by the presence of distracting stimuli (visual, auditory, tactile, etc.) in the background or foreground that contribute to the complexity. The perception of moving stimuli, for older adults, is both less effective and less efficient in tasks aimed at the detection of small target movement/change such as those found on dials and controls (Kline, D. W., & Scialfa, 1997). Furthermore, deficits in central, para-central, and peripheral visual field can pose different demands on vision, resulting in different search strategies related eye movements (Coeckelbergh, Cornelissen, Brouwer, & Kooijam, 2002).

Older adults tend to exhibit a greater degree of difficulty with visual search tasks, especially when the number of items to be searched increases (Kline, D. W., & Scialfa, 1997). Older adults have a propensity for longer visual reaction times, especially in cases where attention is divided. Furthermore, this population segment experiences difficulties ignoring extraneous information, or background noise (Schieber, 1994). Research also suggests that visual search is slower and less effective for older adults due to a shrinking of the useful field of view to which attention can be simultaneously allocated. The size of the useful field of view, for older adults, is especially susceptible to context related factors, such as complexity and cognitive task load (Schieber, 1994).

Summary

Studies have demonstrated that ophthalmic patients are at high risk for decreased functional status and quality of life (Parrish et al., 1997; Scott, Schein, Feuer, Folstein, & Bandeen-Roche, 2001; Scott, Schein, West, Bandeen-Roche, Enger, & Folstein, 1994). Patients' functional statuses and qualities of life may be improved by interventions that increase visual function, such as surgery to repair retinal detachment or remove epiretinal membrane (Scott, Smiddy, Feuer, & Merikansky, 1998) and surgery to remove cataract (Applegate et al., 1987; Brenner, Curbow, Javitt, Legro, & Sommer, 1993; Donderi & Murphy, 1983; Javitt, Brenner, Curbow, Legro, & Street, 1993; Steinberg et al., 1994). In addition, functional status and quality of life may be improved by interventions, such as low vision devices and services, which permit patients to use their remaining vision more effectively (Scott, Smiddy, Schiffman, Feuer, & Pappas, 1999). In addition, as the proportion of older adults multiplies, the number of individuals experiencing some degree of vision dysfunction that normally occurs with age has created an increased

demand for information technology that affords use despite the dysfunction. Prior studies have demonstrated the effect of low vision interventions on objective task-specific measures of functional abilities such as reading speed, reading duration, and ability to read a certain print size (Nilsson, U. L., 1990; Nilsson, U. L., & Nilsson, S. E. G., 1986; Rosenberg, Faye, Fischer, & Budicks, 1989; Sloan, 1968).

However, historically there has been little data available concerning the abilities of people with visual impairments to use computers. It has been even more challenging to find data concerning how modifications of graphical user interface features may increase accessibility of computers to patients with visual impairments. An exception to this is the research agenda established by Jacko and colleagues (see Jacko, Moloney, et al., 2005; Jacko, Scott et al., 2003; Scott et al., 2002a,b; Scott, Jacko, et al., 2006; and Table 43.1 for examples that has yielded systematically derived HCI performance thresholds for users according to their ocular profile (diagnoses and functional ability). Table 43.1 summarizes the research products resulting from this effort, chronologically organized by the ocular pathology investigated, the age of the users involved, the GUI interaction investigated and the corresponding specific interface feature in question. It goes without saying that the rapid proliferation of visual displays beyond the desktop to handheld and wearable computers and other mobile devices, such as cellular telephones, increase the importance of graphical user interface innovations that facilitate efficient and rewarding usage by people with visual impairments.

HIGHLIGHTING TECHNOLOGICAL ADVANCEMENTS IN HCI RESEARCH

With nearly every aspect of today's society involving some type of computer technology, there is an ever-growing need to understand how individuals with visual impairments can access technology. Currently, without special modifications, the typical PC poses several challenges to users who experience limited bandwidth to their visual sensory channels, as well as other perceptual impairments. As a result, the HCI research community is placing great emphasis on the design of universally acceptable technologies. According to Stephanidis, Salvendy, Akoumianakis, Bevan, et al. (1998), "Universal access in the Information Society signifies the right of all citizens to obtain equitable access to, and maintain effective interaction with, a community-wide pool of information resources and artifacts" (p. 6). *Accessibility* has been a term traditionally associated with elderly individuals, individuals with disabilities and others who possess special needs (Stephanidis, Salvendy, Akoumianakis, Arnold, et al., 1999). However, because of the current influx of new technologies into the market, the population of users who may possess special needs is growing. As a result, accessibility has taken on a more comprehensive connotation. This connotation implies that all individuals with varying levels of abilities, skills, requirements, and preferences be able to access information technologies (Stephanidis, Salvendy, Akoumianakis, Arnold, et al., 1999). Universal access also implies more than just adding features to existing technologies. Rather, the concept of universal access emphasizes that accessibility be incorporated directly into the design (Stephanidis, Salvendy, Akoumianakis, Bevan, et al., 1998). Perceptual and adaptive interfaces are two ideal examples of how universal accessibility can be achieved.

Perceptual Interfaces

The concept of perceptual design describes a perspective of design that defines interactions in terms of human perceptual capabilities. In a sense, it strives to humanize interaction. The design of perceptual interfaces adheres to the idea that lessons learned from psychological research about perception can be applied to interface design (Reeves & Nass, 2000). In adopting the concept of perceptual design, several opportunities surface for the creation of innovative perceptual user interfaces. Interactions with these interfaces can be described in terms of three particular human perceptual capabilities; chemical senses (e.g., taste and olfaction), cutaneous senses (e.g., skin and receptors), and vision and hearing (Reeves & Nass, 2000). Although commonly used computer technology limits the effectiveness of chemical senses, the technology can be extended to incorporate the cutaneous, visual, and hearing senses.

In terms of vision, research has focused on topics, such as visual mechanics, color, brightness and contrast, objects and forms, depth, size, and movement. Hearing research includes psychophysical factors such as loudness, pitch, timbre, and sound localization; physiological mechanisms such as the auditory components of the ear, and the neural activity associated with hearing; and the perception of speech such as units of speech and the mechanics of word recognition (Reeves & Nass, 2000). With respect to the cutaneous senses, augmented GUIs with haptic feedback have been around since the early 1990s. Akamatsu and Sate conducted the first research with a haptic mouse that produced haptic feedback via fingertips and force feedback via controlled friction (as cited in Oakley, McGee, Brewster, & Gray, 2000). Engle, Goossens, and Haakma found that directional two degrees of freedom force feedback improved speed and error rates in a targeting task (as cited in Oakley, McGee, Brewster, & Gray, 2000).

The strength of perceptual user interfaces comes from the ability of designers to combine an understanding of natural human capabilities with computer input/output devices, and machine perception and reasoning (Turk & Robertson, 2000). General examples of how capabilities can be combined with technology include speech and sound recognition and generation, computer vision, graphical animation, touch-based sensing and feedback, and user modeling (Turk & Robertson, 2000).

From an applied research standpoint, the concepts of perceptual interfaces are housed within multimedia and multimodal interfaces. Both multimedia and multimodal interfaces offer increased accessibility to technologies for individuals with perceptual impairments. Distinctions can be drawn between perceptual, multimedia, and multimodal interfaces, shown in Fig. 43.2. Perceptual interfaces prescribe human-like perceptual capabilities to the computer. Multimedia and multimodal interfaces can be considered applied extensions of this concept. Multimedia interfaces elicit perceptual and cognitive skills to

TABLE 43.1. Visualization of the Breadth and Depth of the Investigations, To-Date, for the Derivation of HCI Performance Thresholds for Users According to Their Ocular Profile (Diagnoses and Functional Ability). A Shaded Cell Indicates that a Given Topic is Addressed by the Corresponding Investigation, Listed in the Leftmost Column

Investigation		Ocular Pathology				Age		GUI Interaction								GUI Interface Feature						
Year	Authors	None	AMD	Diabetic Retinopathy	Other	18-55 years	55+ years	Visual Search	Icon Selection	Cursor Movement	Drag & Drop	Menu Selection	Distract-ers	Icon Size	Set Size	Back-ground	Visual Profiles	Tactile Cues	Auditory Cues	Text Size	Desk-top PC	Hand-held PC
1998	Jacko & Scars		■	■	■	■	■	■	■				■	■	■	■	■				■	
1999	Jacko		■	■	■	■	■	■	■				■	■	■	■	■				■	
1999	Jacko, Dixon et al.		■	■	■	■	■	■	■				■	■	■	■	■				■	
2000	Jacko,Barreto et al.		■	■	■	■	■	■	■				■	■	■	■	■				■	
2000	Jacko, Rosa et al.		■				■	■		■			■				■				■	
2001	Jacko et al.		■				■	■	■				■	■	■	■	■				■	
2002[a]	Scott, Feuer & Jacko		■				■	■	■				■	■	■	■	■				■	
2002[b]	Scott, Feuer & Jacko		■				■	■	■				■	■	■	■	■				■	
2002	Jacko,Barreto et al.		■				■	■	■				■	■	■	■	■				■	
2002	Vitense,Jacko et al.		■				■				■						■	■	■		■	
2003	Jacko, Vitense & Scott	■				■		■			■		■					■	■		■	
2003	Vitense, Jacko & Emery	■				■		■			■		■					■	■		■	
2003	Emery,Edwards et	■					■				■							■	■		■	
2003	Jacko, Scott et al.		■				■				■						■	■	■		■	
2004	Jacko,Barnard et al.		■				■				■						■	■	■		■	
2004	Jacko, Emery et al.	■					■				■							■	■		■	
2004	Edwards,Barnard et al.			■		■	■	■				■	■	■		■	■	■	■	■	■	
2005	Edwards,Barnard et al.			■		■	■	■				■	■	■		■	■	■	■	■	■	
2005	Jacko,Moloney et al.		■				■				■						■	■	■		■	
2005	Moloney,Leonard et al.		■				■				■						■	■	■		■	
2005	Leonard, Edwards & Jacko			■		■	■	■				■	■	■		■	■	■	■	■	■	
2005	Leonard, Jacko & Pizzimenti		■					■	■	■	■		■	■	■	■	■		■	■		■
2006	Scott et al.		■				■				■						■	■	■		■	
2006	Moloney, Shi et al.		■				■				■						■	■	■		■	

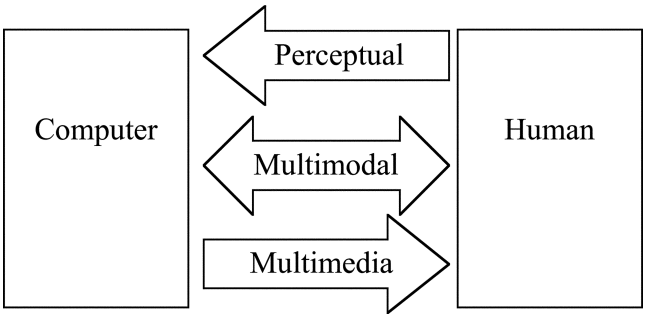


FIGURE 43.2. Perceptual, multimodal, and multimedia interfaces (flow of information).

interpret information presented to the user. Whereas multimodal interfaces use multiple modalities for the HCI. Multimedia interfaces focus on the media while multimodal interfaces focus on the human perceptual channels (Turk & Robertson, 2000). The strength and capabilities of multimedia and multimodal interfaces with respect to individuals with perceptual impairments are described in more depth in the next two sections.

Multimedia interfaces. Multimedia interfaces have grown from the need to display diverse forms of information in a flexible and interactive way. Multimedia can be simply defined as computer-controlled interactive presentations (Chignell & Waterworth, 1997). The broadness of this definition directly corresponds to the broadness of the field of multimedia research. Chapter 12 in this *Handbook* contains a comprehensive discussion of multimedia including topics dealing with the cognitive implications related to multimedia, selecting media for the message, navigation and interaction and evaluation of multimedia. However, for the purpose of this chapter, the discussion of multimedia is limited to a brief overview of types and potential strengths of multimedia as they relate to enhancing accessibility to information technologies for individuals with perceptual impairments.

There are three approaches to multimedia: performance, presentation, and document (Chignell & Waterworth, 1997). In the performance approach, multimedia is a kind of theatrical play that is conveyed through "actors." The timing of the actors' performances is orchestrated in an effort to entertain and educate (Waterworth & Chignell, 1997). Presentation multimedia is a modern version of slide shows, where video clips and animation enhance a sequence of slides. The goal of the presentation approach is to convey ideas to the user (Chignell & Waterworth, 1997). Lastly, the document approach focuses on text and ideas. It can be thought of as an enhanced document that elaborates ideas in the text. All of these approaches provide additional opportunities to convey perceptual information to the user.

Multimedia allows communication between users and computers in a sensory manner. As such, the essential aspect of designing multimedia interfaces is selecting the media and modalities. Information can be taken from one modality and presented in another modality (Chignell & Waterworth, 1997). For instance, data presented visually could be converted and displayed audibly. For instance, multimedia could be used to enhance a GUI for a user with a hearing impairment. Information that would be commonly conveyed via auditory feedback could be provided visually. An example of this would be to have an icon display when an error has been committed, rather than the traditional beep sounding. Other examples of how multiple modalities can enhance accessibility to information technologies are discussed in the multimodal section.

The potential strength of multimedia interfaces comes from its ability to use images, text, and animation to connect with users. However, in order to facilitate that type of exchange, much consideration must be paid to users' sensations and perceptions via the auditory, speech, and visual channels. For instance when designing graphical images, basic knowledge of color vision is needed to ensure that colors can be discriminated, foreground can be separated from background, highlights will attract attention, and grouping of objects is apparent (Gillan, 1998). An understanding of human sensation and perception is necessary if perpetual interfaces are to reach their full potentials. When perceptual interfaces are used, however, they provide individuals with perpetual impairments, such as those related to the visual channel, alternative modalities to access technology. Multimodal interfaces are a second type of perceptual interface that provide this same benefit.

Multimodal interfaces. Multimodal interfaces, as they are discussed in this chapter, are interfaces that support a wide range of perceptual capabilities (e.g., auditory, speech, and visual) as a means to facilitate human interaction with computers. Multimodal interfaces are discussed in a much broader sense (e.g., development of multimodal processing) in another chapter in this handbook.

With the growing complexity of technology and applications, a single modality no longer permits users to interact effectively across all tasks and environments (Oviatt, Cohen, Suhm, et al., 2002). The strength of a multimodal design is its ability to allow users the freedom to use a combination of modalities or the best modality for their needs. These interfaces make the most effective use of the variety of human sensory channels, alone and in combination. Ultimately, multimodal interfaces offer expanded accessibility of computing and promote new forms of computing not previously available to individuals with perceptual impairments.

The development and application of multimodal interfaces for making technologies accessible to users with visual impairments is growing. Some forms of unimodal, bimodal, and trimodal feedback within multimodal interfaces have been investigated and their advantages have been documented for users with visual impairments (Jacko, Emery, et al., 2004; Jacko, Barnard, et al., 2004; Jacko, Moloney, et al., 2005; Jacko, Scott, Sainfort, et al., 2003). Since most information presented on a GUI is visual, there is great interest in the research community to find alternative ways of displaying this information. One of the common approaches to conveying visual information in a nonvisual way is through the use of the tactile modality. Specific research has been conducted in the realm of tactile displays with respect to visual impairments. Not only can tactile displays provide information regarding a graphic's identity but also the depth, location, and perception of its purpose. The use of tactile systems can also provide navigational information. Research has

shown that tactile output of directional information offers support to the blind as they explore images (Kurze, 1998). Research has also been conducted in the area of movable dynamic tactile displays that present information to one or several fingertips in a Braille type manner. The Braille dots move in a wave of lifted and lowered series of pins (Fricke & Baehring, 1994). Along the same line of research, the use of a bidimensional single cell Braille display combined with a standard Braille cell has been evaluated. Although initial research found this new combined device is not an improvement over a standard stand-alone Braille display, continued research is yielding improvement (Ramstein, 1996). Tactile output via force feedback has been looked to as a means of conveying numerical information. Yu, Ramloll, and Brewster (2000) worked on a system that converts data typically displayed visually into haptic and auditory output. Since data visualization techniques are not appropriate for blind people or people with visual impairments, this system translated graphs into friction and textured surfaces along with auditory feedback.

Another common approach to converting visual information in a nonvisual way is through the use of the speech modality. Speech recognition systems serve as an alternative modality for users and computers to interact. These systems recognize human speech and translate it into commands or words understood by the computer. Chapter 34 in this handbook offers an introduction to the technology of speech recognition systems and design issues associated with incorporating speech recognition into applications. However, this chapter's discussion of speech recognition systems concentrates on its use related to individuals with visual impairments. The integration of speech input and output into applications offers an alternative to purely graphical environments (Yankelovich, Levow, & Marx, 1995). This type of technology-driven design allows applications to be suited to a wide variety of individuals with disabilities (Danis & Karat, 1995). For instance, individuals with visual impairments can use a computer solely by voice activation.

Speech recognition systems are traditionally associated with the concept of dictation. Products such as Dragon Systems, Inc. Dragon Naturally Speaking offer a line of speech recognition products for dictation. Some specific packages are geared toward particular professions, such as medical and legal (Cunningham & Coombs, 1997). Other common dictation systems are IBM ViaVoice, Lernout & Hauspie Voice Xpress, and Philips Speech Processing FreeSpeech98. Along the lines of recognition engines, Verbex Voice Systems and SRI Corp.'s DECIPHER offer continuous speech recognition technology. Microsoft's Whisper provides speaker-independent speech recognition with online adaptation, noise robustness, and dynamic vocabularies and grammars (Huang, Acero, Alleva, et al., 1995). Speech driven menu navigation systems, for instance Command Corp. Inc., IN CUBE Voice Command for window navigation, have also been developed (Karshmer, Brawner, & Reiswig, 1994; Karshmer, Ogden, Brawner, Kaugars, & Reiswig, 1994).

Often, circumstances may affect more than one sensory channel, and individuals may experience multiple impairments. In addition to providing heightened challenges to interaction, this impacts the efficacy of perceptual interface in meeting users' needs. For example, many disabilities are associated to some degree with speech degeneration, and therefore speech recognition systems may not fully accommodate a user's needs (Rampp, 1979). Older adults commonly experience multiple impairments, as all the sensory channels normally decline with age, and are more susceptible to conditions, such as stroke, which can cause reduced speech, vision, and motor skills. For cases in which speech is degraded, speech recognition systems must be sensitive enough to adapt to impaired speech. Regardless of what caused the speech impairment, devices and techniques can be applied to augment the communicative abilities of individuals who experience difficulty speaking in an understandable manner.

Augmentative and alternative communication (AAC) is a field of study concerned with providing such devices and techniques. McCoy, Demasco, Pennington, and Badman (1997) described a prototype system aimed at users with cognitive impairments. This prototype is designed to aid communication and provide language intervention benefits across several user populations. An iconic language approach has been applied to aid individuals with significant speech and multiple impairments (SSMI). Research has been conducted on the use of icon language design based on the theory of icon algebra (Chang, 1990) and the theory of conceptual dependency (Schank, 1972). These methodologies are then used in interactive interface design (Albacete, Chang, & Polese, 1994). Individuals with SSMI also commonly have difficulty with word processing. Using an animated graphical display, phoneme probabilities of speech can be more easily isolated and recognized. This allows users the opportunity to interpret their speech rather than forcing the computer to do it automatically (Roy & Pentland, 1998). The more feedback and modalities users are provided, the more efficiently they can interact with speech recognition by assisting the computer to interpret their speech correctly. Speech recognition systems must also take into account the possible extent of cognitive burden placed on the user. If voice is used to navigate through a GUI, certain prosodic features (e.g., pauses) of the user's speech, resulting from a high cognitive load, may affect performance (Baca, 1998). Additional challenges associated with the implementation of speech recognition systems are discussed in this handbook as well. Overall, these systems need to account for situations when users with visual impairment experience varying levels of hearing, speech, and cognitive abilities.

The auditory channel serves not only to convey information, but also to receive information. The use of auditory feedback is extremely useful to many computer users. Ongoing research has looked at the use of bidirectional sound as a standard element of an interface environment. A prototype named the Voice Enabled Reading Assistant (VERA), written using an Aural-Oral User Interface (A-OUI) model, was developed to provide bidirectional sound. The A-OUI model captures qualities and functions of plain text files needed for user interfaces to present information to the auditory, visual, and tactile senses. The use of the VERA prototype can be applied to many types of office and Internet text and data. Similar speech-enabled products are Emacspeak and IBM VoiceType Simply Speaking Gold (Ryder & Ghose, 1999).

Much work with respect to audio feedback has focused on conveying information from a GUI. For instance, research conducted by Darvishi and colleagues (Darvishi, Guggiana, et al., 1994; Darvishi, Munteanu, et al., 1994) looked at mapping GUIs into auditory domains through impact sounds based on

physical modeling. Ultimately, these sounds are used to convey information to the user regarding the objects in a GUI environment. Audio feedback has also been used to provide visually impaired users with a sense of depth perception, by varying the location of the sound sources in a three-dimensional environment. Because depth perception is a function of vision, any cues that can be conveyed though other modalities are vital. This is particularly crucial when working with three-dimensional computer applications, such as a CAD package (Mereu & Kazman, 1996). The use of auditory feedback has also been studied with regard to enhancing synthesis speech output. Through the development of spatial audio processing systems, a greater benefit of synthesis speech was achieved (Crispien, Würz, & Weber, 1994). IBM's Screen Reader/2 also uses synthesis speech to make GUIs accessible to those who are visually impaired or blind, by converting screen information to speech or Braille. The users are continuously kept informed of screen activity and cursor movement. Reading typed characters, words and sentences are features that can be made automatic. This software aids in the use of windows, menus, dialog boxes and other controls (Thatcher, 1994). Microsoft Corporation's Whistler, a trainable, text-to-speech system that produces synthetic speech, sounds very natural by reproducing the characteristics of original speech (Huang, Acero, Adcock, et al., 1996). WRITE:OUTLOUD and OutSPOKEN are two additional text-to-speech products commonly used (Friedlander, 1997). Regardless of what product is used, text-to-speech software enables information to be collected and utilized in a more rapid manner.

In addition to the benefits already discussed that enable users with visual impairments to access technology, multimodal interfaces also provide superior support of error handling. More specifically multimodal interfaces have been shown to have the ability to avoid errors and recover more efficiently from errors when they do occur. This is a very important element of universal accessibility. Enabling users to use a technology includes not only being able to access the technology but also the ability to use it in an efficient manner. Reducing or avoiding errors is a major key to improving efficiency. A multimodal interface provides better error handling than a unimodal interface for several reasons. The following are factors, which represent reasons why a multimodal interface may be better at error avoidance.

- Users will select the input mode considered less error prone for a particular context, which is assumed to lead to error avoidance.
- The ability to use several modalities permits users the flexibility to leverage their strengths by using the most appropriate modality. It is possible that the more comfortable the users, the less likely they are to make errors.
- Users tend to switch modes after systems errors. For instance, users are likely to switch input modes when they encounter a system recognition error.
- Users are less frustrated with error when interacting within a multimodal interface, even when errors are as frequent as in a unimodal interface. The reduction in frustration may be the result of the user feeling as thought they have more control over the system because they can switch modes (Oviatt, 1999).

Ultimately, multimodal systems aid in reducing errors (error handling) because they offer parallel or duplicated functionality that allow users to accomplish the task using one of several modalities (Oviatt & Cohen, 2000). These benefits stem from the user-centered and system-centered design perspectives from which multimodal interfaces are built (Oviatt & Cohen, 2000; Oviatt, Cohen, Suhm, et al., 2002). Like multimodal interfaces, adaptive interfaces, also benefit from a user-centered perspective.

Adaptive Interfaces

Adaptive interfaces have great potential for accommodating a wide range of users in a variety of work contexts. As a result, much research has been conducted on the design and implementation of adaptive interfaces. An example of early research in the field of adaptive GUI is illustrated by the work of Mynatt and Weber (1994) with the Mercator project, and the GUIB (Textual and Graphical User Interfaces for Blind People) project (Petrie, Morley, & Weber, 1995). Both of these projects focused on making environment-level adaptations to GUIs in order to make them more accessible (Stephanidis, 2001a). Mercator interfaces model the graphical objects and their hierarchical relationships. The model serves to predict a user's interaction (Edwards, W. K., & Mynatt, 1994; Edwards, W. K., Mynatt, & Stockton, 1994). The goal is to provide visually impaired users with an interface that is more accessible. By better understanding how low vision users interact with a computer, interfaces can be designed more effectively. Outputs such as synthetic and digitized speech, refreshable Braille, and nonspeech sounds also make an interface easier to use. Auditory icons and earcons are two predominate areas of nonspeech sound research. Auditory icons, developed by Gaver (1989), are everyday sounds that occur in the world mapped to the computer world. Gaver first applied the concept of auditory icons to SonicFinder. Earcons are a second method for employing nonspeech audio in GUIs. Earcons, developed by Blattner, Sumikawa, and Greenberg (1989), are audio messages that provide the user with information about objects or operations of the computer. The first version of GUIB adapted the GUI by combining Braille speech and nonspeech audio together to construct a nonvisual interface so that users who are blind can access the GUI (Emiliani, 2001).

Configurable interface designs based on user models have been heavily researched. The user models have been defined with respect to visual, cognitive, motor, and other abilities. With these models, custom computer systems, including both hardware and software, can be created (McMillan & Wisniewski, 1994). Semantic abstraction of user interaction, named abstract widgets, is another modeling approach that provides great flexibility. Abstract widgets separate the user interface from the application functionality. This allows users to interact with interfaces as they choose, independently from their environment (Kawai, Aida, & Saito, 1996). The use of adaptation determinates, constituents, goals, and rules are yet another approach of an adaptation strategy. This strategy is based on the fact that these important attributes, which categorize adaptation, can be used to formulate adaptation rules. These adaptation rules, in turn, assist the development of intelligent user interfaces. This approach can be customized to the requirements of different application domains

and user groups (Stephanidis, Karagiannidis, & Koumpis, 1997). Approaches such as described by the Pervasive Accessible Technology (PAT) allow individuals with disabilities to use standard interface devices that adapt to the user in order to communicate with information technology infrastructures (Paciello, 1996). Based on an individual's disability, the implementation of a User Interface Management System (UIMS) model provides the versatility needed to adapt interfaces to individuals. The selection of input devices, presentation of information on the screen, and choice in selection/activation method, can all be adapted to fit specific user needs (Bühler, Heck, & Wallbruch, 1994). Current research focuses on operationally reliable software infrastructures that support alternative physical realizations through the abstractions of objects. More specifically, systems such as Active X(by Microsoft and JavaBeans(by SunSoft represent component-ware technology (Stephanidis, 2001b). These systems represent a mainstream effort to provide technological structures that provide more adequate support for accessibility. They are also two examples of currently available tools that can be used to generate code for the adaptation of various interface components.

Some hypermedia systems research has focused on adaptive hypermedia applications. These adaptive hypermedia systems keep track of evolving aspects of the user, such as preferences and domain knowledge. This information is stored to create a user model, which in turn is the basis for user interface adaptation (De Bra, Houben, & Wu, 1999). Adaptive hypermedia can also be designed based on task models. In that the latter case, task models are as the basis from which hypermedia system are developed. These task models support the design and development of hypermedia. Different task models are associated with different types of users (Paternó & Mancini, 1998). Task models reflect the user's view of the activities to be performed. In addition to personalizing the content of hypermedia systems, adaptive navigation support has also been researched. Prototype systems have been developed to demonstrate how different navigational possibilities can be made available based on a user model (Pilar da Silva, Van Durm, Duval, & Oliviè, 1998).

Another example of adapting interfaces to users is through the use of the EZ Access protocols developed by the Trace Center at the University of Wisconsin-Madison. EZ Access protocols are a set of techniques that modify an interface to fit a specific user's need. These protocols work across disabilities and with a range of products. The most common use of EZ Access protocols is in touchscreen kiosks (Vanderheiden, 1998).

While research and development in the areas of perceptual and adaptive interfaces, like those examples discussed, have gained substantial attention in generating potential solutions for individuals with visual sensory impairments, the majority of work and realized products are positioned in two areas. The most readily available products and the majority of the research have been somewhat narrowly focused on (a) the magnification of screen elements (Fraser & Gutwin, 2000) and (b) the accessibility of text (Craven, 2003). Comparatively, less has been accomplished in terms of critical aspects of the graphical user interface as has been done with magnification and vocalization of text. While the solutions generated have afforded access to both those individuals who are blind and those who retain a range of visual impairment, they tend to be one size fits all solutions, which entirely abandon the visual sensory channel (as in the case of Braille interfaces or Screen Readers), or focus only on the augmentation of the visual channel (e.g., magnification, increased contrast, increased text size, etc.), and very few are multimodal, or perceptual. Table 43.2 summarizes a selection of popular assistive devices. Following this table, a discussion of research that has evaluated the efficacy of such approaches to accessibility is presented.

TABLE 43.2. Popular Assistive Devices

Competitor	Product Overview	Key Limitation(s)
Screen Reader 	Software program that reads to the user elements that appear on an interface via synthesized voice. The program reads left to right, starting at the very top of the screen. When an image is encountered the program reads the associated ALT text	Solution abandons any remaining vision the user has, using only their auditory ability Efficacy depends on the organization of the interface (e.g., anything not modeled in a left to right organization is not compatible)
Braille Display 	Similar to the screen readers, but gives the reader the information via tactile cues (Braille characters)	Same limitations as a screen reader, plus the user has to learn Braille, which is not likely if they have residual vision, and are losing vision later in life
Screen Magnifiers 	Physical device or software program that enlarges the entire screen image Software programs, such as Zoomtext, developed by Ai squared, allow for "adaptable" magnification of the interface, and some versions incorporate a screen reader	A 'one size-fits all' solution, that is not adaptable between users For people with visual impairments, magnification is not always the most effective strategy (especially with obstructed visual fields)

Kline and Glinert (1995) presented UnWindows V1, a set of interface tools to support selective magnification of window area, and tracking the location of the mouse pointer on the display screen. The authors noted, "Magnification is one method commonly employed to help low vision users deal with the small type fonts, illustrations, and icons present in much of today's printed media and computer displays" (Kline & Glinert, 1995, p. 2). Key components of the UnWindows system included (a) a dynamic magnifier to compensate for the loss of global context imposed by static magnification and changing display content and (b) Visual and aural feedback to aid the users in locating the mouse pointer. Kline and Glinert placed emphasis on the problematic nature of visual tracking in the presence of a screen densely populated with icons and windows. Interestingly, they received mixed reaction to their interface by users with and without visual impairment, especially in terms of the auditory feedback provided whenever the mouse pointer entered a new window (users found this annoying). While no formal empirical testing was performed in relation to UnWindows, questions surface as to the effectiveness of nonvisual, multimodal feedback in a complex display.

A usability review of currently available technologies for the conversion of GUI technology for use by individuals who were blind or possessed low vision, the ability of magnification, synthetic speech, and Braille were reviewed for their ability provide the respective users 100% access to GUIs on nine test areas (Becker & Lundman, 1998). These tests included

- Installing and configuring the device/software
- Uninstalling the device/software
- Performance reliability/stability
- Program manager to read and manipulate windows, menus and icons
- Word processing based tasks such as opening and saving files, reading and editing text, text attributes, and toolbars
- Spreadsheet tasks such as reading cells, tables figures, and editing data and formulas
- Internet use, including dialing up, accessing World Wide Web pages, navigating with link buttons, sending e-mail and reading graphics
- Screen searching, such as searching for characters, strings, formats, and icons
- Operating start menu, exploring and controlling settings (Becker & Lundman, 1998)

For the assessment of seven magnification programs (synthetic speech and Braille displays are not relevant to individuals with visual impairments who have residual vision), the evaluators comprised of a system engineer, ergonomic engineer, computer science expert, and three individuals with visual impairments.

The use of magnification, as a strategy to afford access to GUIs proved somewhat successful, providing 89% or higher access to GUIs, except for Internet use (84% access) and screen searching (0%; Fortuin & Omata, 2004). It was concluded that the essential problem for the design of interactive systems for users with visual impairments is to (a) to determine what the users need and (b) how to represent the requested information based on key psychological and physical attributes of the user. The result of ineffective assistive technologies is a lack of usable contextual cues for the users to provide feedback in the case of errors; and this translates to large amounts of imposed workload on the user and frustration (Fortuin & Omata, 2004).

In a case study on an English teacher who was having difficulty reading student papers, typing, and proofreading, Whittaker (1998) discovered that magnification was not affording optimal performance. Typically, the authors found that users with visual acuity of 20/40 or better would respond well to simple optical magnification. The authors investigated other visual functioning to find that the individuals had severely diminished contrast sensitivity (13% contrast threshold, with 2% representing normal sensitivity). Furthermore, this individual's visual field was 20 degrees horizontally (180 degrees is normal). Magnification was likely reducing the number of letters viewable simultaneously in the presence of scotoma within the visual field. The author warned that magnifiers and large monitors are not always the most effective solution for users with impaired vision.

Arditi (2004) addressed the reading difficulties of individuals with low vision. According to the author, successfully overcoming this difficulty is accomplished through the exploitation of remaining vision. The easiest way to do this is through magnification, but as shown in this study, it is not a one size fits all solution. Several parameters of the font, including height, stroke, spacing, and serif size, must be selected in a combination that best suits a given user. Arditi presented the prototype and initial user testing of computer-based software that lets a user customize fonts for maximized legibility. Those users studied were able to adjust font to a usable, legible level, to positively impact reading times and the reading acuity.

An in-depth review of accessibility tools aimed at improving interactions of computer users with low vision informed the design of *MouseLupe* (Silva, Regina, & Bellon, 2002). *MouseLupe* simulates a magnifying glass, enabling users to magnify select portions of text or display graphics, inspired by the problematic nature of screen magnification software. The authors suggested that magnification improves the readability of smaller text, but occludes the visible area of the document. Furthermore, graphics that contain text (like most icons), a critical element of the graphical user interface, when enlarged, are difficult to read (for a comprehensive review of magnification tools, see Silva, Regina, & Bellon, 2002).

Several researchers have considered the effect of visual impairments on web browsing (Arditi, 2003; Craven, 2003; Harper, Goble, & Stevens, 2001; Silva, Regina, & Bellon, 2002). Harper, Goble, and Stevens (2001) addressed this problem in terms of "Web Mobility." These authors provided guidelines for movement through and around complex hypermedia environments, such as the web, for users with visual impairments. The problem, according to these authors, is that visual impairment inhibits individuals' ability to efficiently assimilate page structure and visual cues that lead to the following problems:

- Failure to get a feel for the content on the website
- Failure to have a sense for the magnitude of the display or where in a website the interaction takes place

- Disorientation
- Obstacles and distracters such as spacer images, tables, and large images
- Too much complex detail that cannot be resolved
- Frustration

Arditi (2003) observed the problems of web browsing in terms of the allocation of screen space resources. According to the author, conflicts arise in the implementation of web browsing solutions for individuals with low vision including: (a) high magnification requirements; (b) variable typography color, size, and contrasts of the content presented; (c) embedded text messages to augment web images; and (d) accessible web browsing controls (icons, buttons, menus). The author presented a novel approach for effectively using screen resources, providing evidence that the strategic layout of a display is a critical factor to successful interaction. The layout of screen elements was interpreted as more critical than magnification of the screen elements.

Craven (2003) questioned the accessibility of electronic library resources on the World Wide Web for individuals with visual impairments. The results of her study with 20 sighted and 20 visually impaired users revealed the browsing times of those individuals with visual impairments were significantly greater, depending on the design of the website (layout complexity and distracters). Navigation time for the group of users with visual impairments was significantly longer due to visual functioning, but also due to artifacts of assistive technology use in navigation (magnification and screen readers).

Summary

Through technological advances in HCI research, the concepts of perceptual and adaptive interfaces have emerged. These two categories of technologies provide vast opportunities for individuals with perceptual impairments to fully access electronic information. More specifically, multimedia and multimodal systems have been shown to accommodate users of varying abilities. However, while these are the most promising approaches for this population, there still exists a chasm between research and practice. Unfortunately, it is rare to see the knowledge generated through HCI research actually implemented into commercially-available products. Much work still needs to be done, both in terms of advancing knowledge—through empirical HCI research—and in terms of increasing awareness. Translational efforts need to be made to help disseminate this new knowledge into the design of technologies and devices, so that individuals with perceptual (and other forms of) impairment can better leverage the increasingly ubiquitous technological tools used in everyday life. An evidence-based need for improved design of technology—especially for individuals with sensory of perceptual impairments—is needed to help ensure that the new knowledge continually generated through research is ultimately used to improve human interaction with technology.

CONCLUSIONS

To provide a context for the topic of perceptual impairments and computing technologies, Fig. 43.1 demonstrated that, in order to achieve universal access across classes of computing technologies, researchers must be prepared to address several very challenging issues:

1. Establish empirical links between clinical diagnoses and sets of functional capabilities
2. Define the set of functional capabilities required to access information technologies
3. Establish empirical bases for the influence of specific functional capabilities on access to specific classes of technologies

This chapter aimed to establish a basis for addressing such issues by examining, specifically, visual impairment, several specific diagnoses, and the resulting functional abilities. Finally, groundbreaking advancements in perceptual interfaces, multimodal interfaces, multimedia interfaces, and adaptive interfaces was discussed, which can be applied across a variety of classes of technology in order to enhance the perceptual experience of people who possess such impairments.

ACKNOWLEDGMENTS

This chapter was made possible through funding provided to the first author by the National Science Foundation, as well as the Intel Corporation.

References

Age-Related Eye Disease Study Research Group. (2001). A randomized, placebo-controlled, clinical trial of high-dose supplementation with vitamins C and E, beta carotene, and zinc for age-related macular degeneration and vision loss (AREDS Report No. 8). *Archives of ophthalmology, 119*, 1417–1436.

Albacete, P. L., Chang, S. K., & Polese, G. (1994). *Iconic Language Design for People with Significant Speech and Multiple Impairments.* Paper presented at the First Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '94), Marina Del Rey, CA.

Assistive Technologies for Independent Aging: Opportunities and Challenges: Hearing Before the Special Committee on Aging, 108th Cong., 2nd Sess., 145 (2004) (statement of the American Foundation for the Blind).

Applegate, W. B., Miller, S. T., Elam, J. T., Freeman, J. M., Wood, T. O., & Gettlefinger, T. C. (1987). Impact of cataract surgery with lens imiplantation on vision and physical function in elderly patients. *JAMA, 257*, 1064–1066.

Arditi, A. (2002). Web accessibility and low vision. *Aging & Vision, 14*(2), 2–3.

Arditi, A. (2003). Low vision web browsing and allocation of screen space resources. *Investigative Ophthalmology and Visual Science, 44*(5), 2767.

Arditi, A. (2004). Adjustable typography: an approach to enhancing low vision text accessibility. *Ergonomics, 47*(5), 469–482.

Baca, J. (1998). *Comparing effects of navigational interface modalities on speaker prosodics.* Paper presented at the Third Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '98), Marina Del Rey, CA.

Becker, S., & Lundman, D. (1998). *Improving access to computers for blind and visually impaired persons: The development of a test method for usability.* Paper presented at the Telemeatics for the Integration of Disabled and Elderly People (TIDE) 1998 Conference, Helsinki, Finland.

Blattner, M., Sumikawa, D., & Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction, 4*(1), 11–44.

Brenner, M. H., Curbow, B., Javitt, J. C., Legro, M. W., & Sommer, A. (1993). Vision change and quality of life in the elderly: Response to cataract surgery and treatment of other chronic ocular conditions. *Archives of ophthalmology, 111*(5), 680–685.

Brewster, S. A., Wright, P. C., & Edwards, A. D. N. (1994). *The design and evaluation of an auditory-enhanced scrollbar.* Paper presented at the ACM Conference on Human Factors in Computing Systems (CHI), Boston, MA.

Bühler, C., Heck, H., & Wallbruch, R. (1994). *A Uniform Control Interface for Various Electronic Aids.* Paper presented at the 4th international conference on Computers for handicapped persons, Vienna, Austria.

Change, S.-K. (Ed.). (1990). *Principles of visual programming systems*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Chignell, M., & Waterworth, J. (1997). Multimedia. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomic* (pp. 1808–1861). New York, NY: John Wiley & Sons.

Coeckelbergh, T. R. M., Cornelissen, F. W., Brouwer, W. H., & Kooijam, A. C. (2002). The effect of visual field defects on eye movement and practical fitness to drive. *Vision Research, 42*, 669–677.

Colenbrander, A. (1977). Dimensions of visual performance. *Transactions—American Academy of Ophthalmology and Otolaryngology, 83*(2), 332–337.

Craven, J. (2003). Access to electronic resources by visually impaired people. *Information Research, 8*(4), paper no. 156 [Available at: http://informationr.net/ir/8-4/paper156.html].

Crispien, K., Würz, W., & Weber, G. (1994). *Using spatial audio for the enhanced presentation of synthesised speech within screen-readers for blind computer users.* Paper presented at the 4th international conference on Computers for handicapped persons, Vienna, Austria.

Cunningham, C., & Coombs, N. (1997). *Information access and adaptive technology*. Phoenix, AZ: American Council on Education and The Oryx Press.

Danis, C., & Karat, J. (1995). *Technology-driven design of speech recognition systems.* Paper presented at the conference on Designing interactive systems: processes, practices, methods, & techniques, Ann Arbor, MI.

Darvishi, A., Guggiana, V., Munteanu, E., Schauer, H., Motavalli, M., & Rauterberg, M. (1994). *Synthesizing nonspeech sound to support blind and visually impaired computer users.* Paper presented at the ICCHP, Vienna, Austria.

Darvishi, A., Munteanu, E., Guggiana, V., Schauer, H., Motavalli, M., & Rauterberg, M. (1994). *Automatic impact sound generation for using in nonvisual interfaces.* Paper presented at the First Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '94), Marina Del Rey, CA.

De Bra, P., Houben, G.-J., & Wu, H. (1999). *AHAM: A Dexter-based reference model for adaptive hypermedia.* Paper presented at the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: Returning to our diverse roots, Darmstadt, Germany.

Diabetic Retinopathy Study Research Group. (1979). Four risk factors for severe visual loss in diabetic retinopathy: DRS Report 3. *Archives of ophthalmology, 97*, 654–655.

Diabetic Retinopathy Study Research Group. (1981). Photocoagulation treatment of proliferative diabetic retinopathy: clinical application of diabetic retinopathy study (DRS) findings (DRS Report 8). *Archives of ophthalmology, 88*, 583–600.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). *Human-computer interaction* (2nd ed.). New York, NY: Prentice Hall.

Donderi, D. C., & Murphy, S. B. (1983). Predicting activity and satisfaction following cataract surgery. *Journal of Behavioral Medicine, 6*, 313–328.

Early Treatment Diabetic Retinopathy Study Research Group. (1995). Focal photocoagulation treatment of diabetic macular edema (ETDRS Report 19). *Archives of Ophthalmology, 113*, 1144–1155.

Edwards, P. J., Barnard, L., Emery, V. K., Yi, J., Moloney, K. P., Kongnakorn, T., et al. (2005). Understanding users with diabetic retinopathy: Factors that affect performance in a menu selection task. *Behavior and Information Technology, 24*(3), 175–186.

Edwards, P. J., Barnard, L., Emery, V. K., Yi, J., Moloney, K. P., Kongnakorn, T., et al. (2004, October). Strategic design for users with diabetic retinopathy: Factors influencing performance in a menu-selection task. *Proceedings of the Sixth International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS'04*), Atlanta, GA, 118–125.

Edwards, W. K., & Mynatt, E. D. (1994). *An architecture for transforming graphical interfaces.* Paper presented at the ACM Conference on User Interface Software and Technology (UIST '94), Marina Del Rey, CA.

Edwards, W. K., Mynatt, E. D., & Stockton, K. (1994). *Providing access to graphical interfaces, not graphical screens.* Paper presented at the ACM Conference on Assistive and Enabling Technology (ASSETS '94), Marina Del Rey, CA.

Emery, V. K., Edwards, P. J., Jacko, J. A., Moloney, K. P., Barnard, L., Kongnakorn, T., et al. (2003, November). Toward achieving universal usability for older adults through multimodal feedback. *Proceedings of the 2003 ACM Conference on Universal Usability,* Vancouver, British Columbia, 46–53.

Emiliani, P. L. (2001). Special needs and enabling technologies: An evolving approach to accessibility. In C. Stephanidis (Ed.), *User interfaces for all—concepts, methods and tools* (pp. 97–113). Mahwah, NJ: Lawrence Erlbaum Associates.

The Eye Diseases Prevalence Research Group. (2004a). Causes and prevalence of visual impairment among adults in the United States. *Archives of Ophthalmology, 122*, 477–485.

The Eye Diseases Prevalence Research Group. (2004b). Prevalence of age-related macular degeneration in the United States. *Archives of Ophthalmology, 122*, 564–572.

The Eye Diseases Prevalence Research Group. (2004c). The prevalence of diabetic retinopathy among adults in the United States. *Archives of Ophthalmology, 122*, 552–563.

Fletcher, D. C. (Ed.). (1999). *Low vision rehabilitation: Caring for the whole person*. Hong Kong: Oxford University Press.

Fletcher, D. C., Schuchard, R., Livingstone, C., Crane, W., & Hu, S. (1994). Scanning laser ophthalmoscope macular perimetry and applications for low vision rehabilitation clinicians. *Ophthalmology Clinics of North America, 7*, 257–265.

Fortuin, F. T., & Omata, S. (2004). Designing universal user interfaces—The application of universal design rules to eliminate information barriers for the visually impaired and the elderly. Retrieved May 19, 2004, from http://www.visionconnection.org

Fraser, J., & Gutwin, C. (2000). *A framework of assistive pointers for low vision users.* Paper presented at the ACM Conference on Assistive Technologies, Arlington, VA.

Fricke, J., & Baehring, H. (1994). *Displaying laterally moving tactile information.* Paper presented at the 4th international conference on Computers for handicapped persons (ICCHP'94), Vienna, Austria.

Friedlander, C. (1997). Speech facilities for the reading disabled. *Communications of the ACM, 40*(3), 24–25.

Gaver, W. (1989). The SonicFinder: An interface that uses auditory icons. *Human Computer Interaction, 4*(1), 67–94.

Gerber, E., & Kirchner, C. (2001, March). Who's surfing? Internet access and computer use by visually impaired youths and adults. *Journal of Visual Impariment & Blindness*, 176–181.

Gillan, D. J. (1998). *The psychology of multimedia: Principles of perception and cognition.* Paper presented at the 1998 ACM Conference on Human Factors in Computing Systems (CHI'98), Los Angeles, CA.

Gragoudas, E. S., Adamis, A. P., Cunningham, E. T., Jr., Feinsod, M., & Guyer, D. R. (2004). Pegaptanib for neovascular age-related macular degeneration. *New England Journal of Medicine, 351*, 2805–2816.

Harper, S., Goble, C., & Stevens, R. (2001). Web mobility guidelines for visually impaired surfers. *Journal of Research and Practice in Information Technology, 33*(1), 30–41.

Huang, X., Acero, A., Adcock, J., Hon, H.-W., Goldsmith, J., Liu, J., et al. (1996). *Whistler: A trainable text-to-speech system.* Paper presented at the Fourth International Conference of Spoken Language Processing (ICSLP '96), Philadelphia, PA.

Huang, X., Acero, A., Alleva, F., Hwang, M.-Y., Jiang, L., & Mahajan, M. (1995). *Microsoft Windows highly intelligent speech recognizer: Whisper.* Paper presented at the International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI.

Jacko, J. A. (1999, August) The importance of clinical diagnoses in the prediction of performance on computer-based tasks for low vision users. *Proceedings of the 9th International Conference on Human-Computer Interaction*, Munich, 787–791.

Jacko, J. A., Dixon, M. A., Rosa, R. H., Jr., Scott, I. U., & Pappas, C. J. (1999). Visual profiles: A critical component of universal access. *ACM Conference on Human Factors in Computing Systems* (*CHI 99*), Pittsburg, PA, 330–337.

Jacko, J. A., Barnard, L., Kongnakorn, T., Moloney, K. P., Edwards, P. J., Emery, V. K., et al. (2004). Isolating the effects of visual impairment: Exporing the effect of AMD on the utility of multimodal feedback. *CHI Letters, 6*(1), 311–318.

Jacko, J. A., Barreto, A. B., Scott, I. U., Chu, J. Y. M., Vitense, H. S., Conway, F. T., et al. (2002). Macular degeneration and visual icon use: Deriving guidelines for improved access. *Univeral Access in the Information Society, 1*, 197–296.

Jacko, J. A., Barreto, A. B., Scott, I. U., Rosa, R. H., Jr., & Pappas, C. J. (2000). Using electroencephalogram to investigate stages of visual search in visually impaired computer users: Preattention and focal attention. *International Journal of Human-Computer Interaction, 12*(1), 135–150.

Jacko, J. A., Emery, V. K., Edwards, P. J., Ashok, M., Barnard, L., Kongnakorn, T., et al. (2004). The effects of multimodal feedback on older adults' task performance given varying levels of computer experience. *Behavior and Information Technology, 23*(4), 247–264.

Jacko, J. A., Moloney, K. P., Kongnakorn, T., Barnard, L., Edwards, P. J., Emery, V. K., et al. (2005). Multimodal feedback as a solution to ocular disease-based user performance decrements in the absence of functional visual loss. *International Journal of Human-Computer Interaction, 18*(2), 183–218.

Jacko, J. A., Dixon, M. A., Rosa, R. H., Scott, I. U., & Pappas, C. J. (1999). Linking visual capabilities of partially sighted computer users to psychomotor task performance. *Proceedings of the 9th International Conference on Human-Computer Interaction*, Munich, Germany, August 22–27, 975–979.

Jacko, J. A., Rosa, R. H., Scott, I. U., Pappas, C. J., & Dixon, M. A. (2000). Visual impairment: The use of visual profiles in evaluations of icon use in computer-based tasks. *International Journal of Human-Computer Interaction, 12*(1), 151–165.

Jacko, J. A., Scott, I. U., Barreto, A. B., Bautsch, H. S., Chu, J. Y. M., & Fain, W. B. (2001, August). *Iconic visual search strategies: A comparison of computer users with AMD versus computer users with normal vision.* Paper presented at the 9th International Conference on Human-Computer Interaction, New Orleans, LA.

Jacko, J. A., Scott, I. U., Sainfort, F., Moloney, K. P., Kongnakorn, T., Zorich, B. S., et al. (2003). Effects of multimodal feedback on the performance of older adults with normal and impaired vision. *Lecture Notes in Computer Science* (*LNCS*)*, 2615*, 3–22.

Jacko, J. A., & Sears, A. (1998). *Designing interfaces for an overlooked user group: Considering the visual profiles of partially sighted users.* Paper presented at the ACM Conference on Assistive Technologies, Marina del Rey, CA.

Jacko, J. A., & Vitense, H. S. (2001). A review and reappraisal of information technologies within a conceptual framework for individuals with disabilities. *Universal Access in the Information Society* (*UAIS*)*, 1*, 56–76.

Jacko, J. A., Vitense, H. S., & Scott, I. U. (2003). Perceptual impairments and computing technologies. In J. A. Jacko & A. Sears (Eds.), *Human-computer interaction handbook* (pp. 504–522). Mahwah, NJ: Lawrence Erlbaum Associates.

Javitt, J. C., Brenner, M. H., Curbow, B., Legro, M. W., & Street, D. A. (1993). Outcomes of cataract surgery. Improvement in visual acuity and subjective visual function after surgery in the first, second, and both eyes. *Archives of ophthalmology, 111*(5), 686–691.

Karshmer, A. I., Brawner, P., & Reiswig, G. (1994). An experimental sound-based hierarchical menu navigation system for visually handicapped use of graphical user interfaces. *Proceedings of the first annual ACM conference on Assistive technologies*, Marina Del Rey, CA, 123–128.

Karshmer, A. I., Ogden, B., Brawner, P., Kaugars, K., & Reiswig, G. (1994). *Adapting graphical user interfaces for use by visually handicapped computer users: Current results and continuing research.* Paper presented at the 4th international conference on Computers for handicapped persons (ICCHP'94), Vienna, Austria.

Kawai, S., Aida, H., & Saito, T. (1996). *Designing interface toolkit with dynamic selectable modality.* Paper presented at the second annual ACM conference on Assistive technologies, Vancouver, Canada.

Kline, D. W., & Scialfa, C. T. (1997). Sensory and perceptual functioning: Basic research and human factor implications. In A. D. Fisk & W. A. Rogers (Eds.), *Handbook of human factors and the older adult* (pp. 27–54). San Diego, CA: Academic Press.

Kline, R. L., & Glinert, E. P. (1995). *Improving GUI accessibility for people with low vision.* Paper presented at the SIGCHI Conference on Human Factors in Computing Systems, Denver, CO.

Kurze, M. (1998). *TGuide: A guidance system for tactile image exploration.* Paper presented at the Third Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '98), Marina Del Rey, CA.

LaPlante, M. P. (1988). Prevalence of conditions causing need for assistance in activities of daily living. In M. P. LaPlante (Ed.), *Data on disability from the National Health Interview Survey, 1983–1985* (p. 3). Washington, DC: National Institute on Disability and Rehabilitation Research.

Lee, H.-W., Legge, G. E., & Oritz, A. (2003). Is word recognition different in central and peripheral vision? *Vision Research, 43*, 2837–2846.

Leonard, V. K., Edwards, P. J., & Jacko, J. A. (2005, September) Informing accessible design through self-reported quality of visual health. *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*, Orlando, FL, 994–998.

Leonard, V. K., Jacko, J. A., & Pizzimenti, J. J. (2005, October) An exploratory investigation of handheld computer interaction for older adults with visual impairments. *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS 2005*), Baltimore, MD, 12–19.

McCoy, K. F., Demasco, P., Pennington, C. A., & Badman, A. L. (1997). *Some interface issues in developing intelligent communication aids for people with disabilities.* Paper presented at the 1997 international conference on Intelligent user interfaces (IUI'97), Orlando, FL.

McMillan, W. W., & Wisniewski, L. (1994). *A rule-based system that suggests computer adaptations for users with special needs.* Paper presented at the First Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '94), Marina Del Rey, CA.

Mehr, E. B., & Freid, A. N. (1975). *Low vision care.* Chicago: Professional Press.

Mereu, S. W., & Kazman, R. (1996). *Audio enhanced 3D interfaces for visually impaired users.* Paper presented at the 1996 ACM Conference on Human Factors in Computing Systems (CHI'96), Vancouver, Canada.

Moloney, K. P., Shi, B., Leonard, V. K., Jacko, J. A., Vidakovic, B., & Sainfort, F. (in press). Leveraging data complexity: Pupillary behavior of older adults with visual impairment during HCI. *Transactions on Computer-Human Interaction* (*TOCHI*).

Mynatt, E. D., & Weber, G. (1994). *Nonvisual presentation of graphical user interfaces: contrasting two approaches.* Paper presented at the 1994 ACM Conference on Human Factors in Computing Systems (CHI'94), Boston, MA.

Nilsson, U. L. (1990). Visual rehabilitation of patients with and without educational training in the use of optical aids and residual vision: a prospective study of patients with advanced age-related macular degeneration. *Clinical Vision Sciences, 6*, 3–10.

Nilsson, U. L., & Nilsson, S. E. G. (1986). Rehabilitation of the visually handicapped with advanced macular degeneration. *Documenta Ophthalmologica, 62*(4), 345–367.

Oakley, I., McGee, M. R., Brewster, S. A., & Gray, P. (2000). Putting the feel in 'look and feel'. *Proceedings of the SIGCHI conference on Human factors in computing systems* Hague, Netherlands, 415–422.

Orr, A. L. (1998). *Issues in aging and vision*. New York: American Foundation for the Blind.

Oviatt, S. (1999). *Mutual disambiguation of recognition errors in a multimodel architecture.* Paper presented at the SIGCHI conference on Human factors in computing systems: The CHI is the limit (CHI '99), Pittsburgh, PA.

Oviatt, S., & Cohen, P. (2000). Multimodal interfaces that process what comes naturally. *Communications of the ACM, 43*(3), 45–53.

Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., & Ferro, D. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions, *Human Computer Interaction*, *15*(4), 263–332.

Paciello, M. G. (1996). Designing for people with disabilities. *Interactions, 3*(1), 15–16.

Parrish, R. K. I., Gedde, S. J., Scott, I. U., Feuer, W. J., Schiffman, J. C., Mangione, C. M., et al. (1997). Visual function and quality of life among patients with glaucoma. *Archives of ophthalmology, 115*(11), 1447–1455.

Paternó, F., & Mancini, C. (1998). *Developing adaptable hypermedia.* Paper presented at the 4th international conference on Intelligent user interfaces (IUI '99), Los Angeles, CA.

Petrie, H., Morley, S., & Weber, G. (1995). *Tactile-based direct manipulation in GUIs for blind users.* Paper presented at the Conference companion on Human factors in computing systems (CHI '95), Denver, CO.

Pilar da Silva, D., Van Durm, R., Duval, E., & Oliviè, H. (1998). *Adaptive navigational facilities in educational hypermedia.* Paper presented at the ninth ACM conference on Hypertext and hypermedia: Links, objects, time and space—Structure in hypermedia systems (HyperText '98), Pittsburgh, PA.

Rampp, D. L. (1979). Hearing and learning disabilities. In L. J. Bradford & W. G. Hardy (Eds.), *Hearing and Hearing Impairment* (pp. 381–389). New York: Grune & Stratton.

Ramstein, C. (1996). *Combining hepatic and Braille technologies: Design issues and pilot study.* Paper presented at the Second Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '96), Vancouver, Canada.

Reeves, B., & Nass, C. (2000). Perceptual user interfaces: Perceptual bandwidth. *Communications of the ACM, 43*(3), 65–70.

Rosenberg, R., Faye, E., Fischer, M., & Budicks, D. (1989). Role of prism relocation in improving visual performance of patients with macular dysfunction. *Optometry and Vision Science, 66*(11), 747–750.

Roy, D., & Pentland, A. (1998). *A phoneme probability display for individuals with hearing disabilities.* Paper presented at the Third Annual International ACM/SIGCAPH Conference on Assistive Technologies (ASSETS '98), Marina Del Rey, CA.

Ryder, J. W., & Ghose, K. (1999). *Multi-sensory browser and editor model.* Paper presented at the 1999 ACM symposium on Applied computing, San Antonio, TX.

Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology, 3*(4), 532–631.

Schieber, F. (1994). *Recent developments in vision, aging and driving: 1988–1994*. Report No. UMTRI-94-26. Ann Arbor, MI: University of Michigan, Transportaion Research Institute.

Scott, I. U., Feuer, W. J., & Jacko, J. A. (2002a). Impact of graphic user interface screen features on computer task accuracy and speed in a cohort of patients with age-related macular degeneration. *American Journal of Ophthalmology, 134*(6), 857–862.

Scott, I. U., Feuer, W. J., & Jacko, J. A. (2002b). Impact of visual function on computer task accuracy and reaction time in a cohort of patients with age-related macular degeneration. *American Journal of Ophthalmology, 133*(3), 350–357.

Scott, I. U., Jacko, J. A., Sainfort, F., Leonard, V. K., Kongnakorn, T., & Moloney, K. P. (2006). The impact of auditory and haptic feedback on computer task performance in patients with Age-related Macular Degeneration and control subjects with no known ocular disease. *Retina, 26*(7), 803–810.

Scott, I. U., Schein, O. D., Feuer, W. J., Folstein, M. F., & Bandeen-Roche, K. (2001). Emotional distress in patients with retinal disease. *American Journal of Ophthalmology, 131*(5), 584–589.

Scott, I. U., Schein, O. D., West, S., Bandeen-Roche, K., Enger, C., & Folstein, M. F. (1994). Functional status and quality of life measurement among ophthalmic patients. *Archives of Ophthalmology, 112*(3), 329–335.

Scott, I. U., Smiddy, W. E., Feuer, W., & Merikansky, A. (1998). Vitreoretinal surgery outcomes: Results of a patient satisfaction/functional status survey. *Ophthalmology, 105*(5), 795–803.

Scott, I. U., Smiddy, W. E., Schiffman, J., Feuer, W. J., & Pappas, C. J. (1999). Quality of life of low-vision patients and the impact of low-vision services. *American Journal of Ophthalmology, 128*(1), 54–62.

Silva, L., & Pereira Bellon, O. R. (2002). A novel application to aid low vision computer users. *Lecture Notes in Computer Science*, 2398, 455–462.

Sloan, L. L. (1968). Reading aids for the partially sighted: Factors which determine success or failure. *Archives of Ophthalmology, 80*(1), 35–38.

Steinberg, E. P., Tielsch, J. M., Schein, O. D., Javitt, J. C., Sharkey, P., Cassard, S. D., et al. (1994). National study of cataract surgery outcomes: Variation in 4-month postoperative outcomes as reflected in multiple outcome measures. *Ophthalmology, 101*(6), 1131–1140.

Stephanidis, C. (2001a). The concept of unified user interfaces. In C. Stephanidis (Ed.), *User interfaces for all: Concepts, methods, and tools* (pp. 371–388). Mahwah, NJ: Lawrence Erlbaum Associates.

Stephanidis, C. (Ed.). (2001b). *User interfaces for all: Concepts, methods, and tools.* Mahwah, NJ: Lawrence Erlbaum Associates.

Stephanidis, C., Karagiannidis, C., & Koumpis, A. (1997). *Decision making in intelligent user interfaces.* Paper presented at the 2nd international conference on Intelligent user interfaces (IUI '97), Orlando, FL.

Stephanidis, C., Salvendy, G., Akoumianakis, D., Arnold, A., Bevan, N., Dardailler, D., et al. (1999). Toward an information society for all: HCI challenges and R&D recommendations. *International Journal of Human-Computer Interaction, 11*(1), 1–28.

Stephanidis, C., Salvendy, G., Akoumianakis, D., Bevan, N., Brewer, J., Emiliani, P. L., et al. (1998). Toward an information society for all: An international research and development agenda. *International Journal of Human-Computer Interaction, 10*(2), 107–134.

Thatcher, J. (1994). *Screen reader/2: access to OS/2 and the graphical user interface.* Paper presented at the first annual ACM conference on Assistive technologies (ASSETS '94), Marina Del Rey, CA.

Turk, M., & Robertson, G. (2000). Perceptual user interfaces. *Communications of the ACM, 43*(3), 32–34.

Vanderheiden, G. C. (1998). Universal design and assistive technology in communication and information technologies: alternatives or complements? *Assistive Technology, 10*(1), 29–36.

Vitense, H. S., Jacko, J. A., & Emery, V. K. (2002). Foundations for improved interaction by individuals with visual impairments through multimodal feedback. *Universal Access in the Information Society* (*UAIS*)*, 2*(1), 76–87.

Vitense, H. S., Jacko, J. A., & Emery, V. K. (2003). Multimodal feedback: An assessment of performance and mental workload. *Ergonomics, 46*(1–3), 58–87.

Waterworth, J. A., & Chignell, M. H. (1997). Multimedia interaction. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 915–946). New York, NY: Elsevier Science.

Whittaker, S. G. (1998). *Choosing assistive devices when computer users have impaired vision.* Paper presented at the Center on Disabilities, Technology, and Persons with Disabilities Conference, Northridge, CA.

Yankelovich, N., Levow, G.-A., & Marx, M. (1995). *Designing SpeechActs: Issues in speech user interfaces.* Paper presented at the SIGCHI conference on Human factors in computing systems, Denver, CO.

Yu, W., Ramloll, R., & Brewster, S. (2000). *Haptic graphs for blind computer users.* Paper presented at the First Workshop on Haptic Human-Computer Interaction, Glasgow, Scotland.

•44•

UNIVERSAL ACCESSIBILITY AND FUNCTIONALLY ILLITERATE POPULATIONS: IMPLICATIONS FOR HCI, DESIGN, AND TESTING

William M. Gribbons
Bentley College

Introduction **872**
Defining the Population 872
Population Size 872
Making the Case for Including Functional Illiteracy 873
Functional illiteracy and healthcare 873
Characteristics of the Population **874**
Functional illiteracy and aging 874
Functional Characteristics of the Learning Disabled 874
Reading 875
Memory 876
Long-term memory 876
Working memory 876
Metacognitive 877
Navigation and Search 877
Interaction skills 877
Topology 878
Depth vs. breadth 878
Navigation aids 878
Search 878
Design Accommodations **879**
Assistive technologies 879
Layered design 879
Universal design 879
Reading 879
Memory 880
Metacognitive 880
Navigation and Search 880
Guidance for Usability Testing 881
Conclusion **881**
References **881**

INTRODUCTION

Over the past decade, the topic of universal accessibility has received close attention in the field of system design. While considerable progress has been made in addressing the needs of the physically disabled and aging, the cognitively disabled have largely been overlooked (Gribbons, 1992, Newell, Carmichael, Gregor, and Alm, 2003). With nearly 45 million adult Americans suffering from the debilitating effects of illiteracy, the Universal Accessibility movement has clearly not been universal. While many have called for action (Dickinson, Eisma, & Gregor, 2003; Shneiderman, 2000), an extensive review of the leading human factors and HCI journals revealed a small number of studies focused on this critical issue. The topic of the less literate and learning disabled is also glaringly omitted from the leading HCI and human factors textbooks.

The research that does exist in literacy and learning disability studies has been confined, for the most part, to the fields of educational psychology, instructional design, and health-care informatics. Here, a rich research tradition dates back many decades. However, most of these findings are limited to the design of the classroom experience, standalone computer-based training systems, and paper-based communication products. Research in noneducational settings for general interface and interaction design is severely limited. Further, most of this work focuses on children and not the adult population.

The ubiquitous nature of the Internet has forced some progress on accommodations for the cognitively disabled. The Web Content Accessibility Guidelines 1.0 (1999) released by the W3C urges designers to accommodate this population. Unfortunately, many emerging guidelines tend to be vague and nonspecific.

This chapter will address the challenge of accommodating functionally illiterate users and/or users with learning disabilities in system design. It will explore the nature of this population and the magnitude of the problem, the relationship between literacy and learning disability, characteristics of the disability, and recommendations for design practice and usability testing. Ultimately, it is hoped that this effort will encourage further research in this critical area and the integration of that work in our profession's educational programs, guidelines, and best practices.

Defining the Population

Part of the challenge of addressing this population is building consensus on the most effective definition of this group. Common labels include "functionally illiterate," "cognitively disabled," and "the learning disabled" (a subset of the cognitively disabled group). An underlying assumption in this work is that the three categories are not mutually exclusive. For the purposes of this chapter, discussion will be limited to functionally illiterate and learning-disabled populations. Functional illiteracy, a lack of document and quantitative literacy needed to function in modern society, was selected since it provides the most comprehensive picture of the total population. This perspective brings to the discussion a very detailed set of demographics and statistics. Unfortunately, this perspective is weak on underlying causes and effective accommodations. Fortunately, the learning-disabled perspective, while focused most often on the school-age population, provides an extensive research base that defines the characteristics of this population, the underlying source of the disability, and possible accommodations. Learning disability is an umbrella term used to describe a wide range of disorders in information processing and it is generally believed that learning disabilities are linked to a dysfunction in the central nervous system. The larger cognitively disabled category was not chosen since it includes a much broader class of disabilities including Down syndrome, autism, emotional disabilities, and Alzheimer's disease. Naturally, these more severe disabilities require more extreme accommodations.

The causes of functional illiteracy are complex and often inextricably intertwined; these causes include a lack of educational attainment, dyslexia, other learning disabilities, and social deprivation. Further complicating matters, many adults with low literacy skills were never diagnosed with a learning disability as children and entered adulthood suffering the effects of the disability without knowing the cause. Consequently, it is all but impossible to characterize whether the cause of adult illiteracy is a product of low educational attainment, an underlying disability, or whether an undiagnosed disability was the cause of an individual dropping out of the educational system. While it may never be possible to determine how many in the lower-literacy population have learning disabilities, it is a safe assumption that the contribution to the population is significant.

Population Size

The current state of functional illiteracy in the United States presents a shameful picture. According to the 1992 National Adult Literacy Survey, some 23%, or nearly 45 million of approximately 200 million adult Americans, function at the lowest level of literacy, "Level 1." Those at Level 1 literacy are, for the most part, able only to read a simple form or understand rudimentary information in a short news article; others are not capable of even this (Kirsch, Jungeblut, Jenkins, & Kolstad, 1993). Within the context of these figures, it is surprising that greater attention has not been directed to this population, since its numbers are larger than all other disability groups combined.

The other literacy levels, two through five, represent progressively higher levels of literacy skills. Not surprisingly, one contributor to functional illiteracy is low educational attainment. The National Adult Literacy Survey states that "adults with relatively few years of education were more likely to perform in the lower literacy levels than those who completed high school or received some postsecondary education. (Kirsch, Jungeblut, Jenkins, and Kolstad, 1993). The National Center for Educational Statistics (2001) reported the following:

> "Between 1972 and 1985, high school completion rates climbed by 2.6 percentage points (from 82.8 percent in 1972 to 85.4 percent in 1985); since 1985, the rate has shown no consistent trend and has fluctuated between 85 and 87 percent. This net increase of about 3 percentage points over 29 years represents slow progress toward improving the national high school completion rates" (High School Completion Rates, Executive Summary). Kirsch, Jungeblut, Jenkins, & Kolstad, 1993.

Given this slow progress, it is highly probable that educational attainment will continue to be a key contributor to lower literacy skills among adults for the foreseeable future.

The consensus of most literacy experts is that people at Levels 1 and 2, the lowest levels, lack the essential skills to function successfully in society. In 2003, this survey was readministered and the results showed no significant improvement in prose and document literacy, with some improvement in quantitative literacy. Cleary the data from these two studies show the problem is large and gains in literacy—like increases in high school completion rates—will come slowly.

Unlike the measure provided by the National Adult Literacy Survey, there is no large-scale study measuring the prevalence of learning disabilities in the general population. There are, however, a range of estimates from a variety of sources. The National Dyslexia Association (n.d.) estimates that 70–80% of the population of students receiving special education services has deficits in reading and that 15–20% of the general population has language-based learning disabilities (how common are language-based learning disabilities simply to document the scope of the problem). The Interagency Committee on Learning Disabilities believes that 5–10% of the population is affected by learning disabilities. The President's Committee on Employment for People with Disabilities found that 10–14% of the adults in the workforce have learning disabilities. Finally, the National Institutes of Health estimated that 15% of the population in the United States has some type of learning disability (National Institute for Literacy, n.d.). Despite the probable link between those with learning disabilities and the larger lower-literacy population, no attempt has been made to validate this connection.

Not surprisingly, functional illiteracy and illiteracy are not problems confined to the United States. There are an estimated 876 million illiterate adults in the world, which represents nearly a quarter of the world's population. In developed nations, it is estimated that there are approximately 100 million functionally illiterate people (Kickbusch, 2001). In India alone, nearly 45% of the adult population is illiterate according to the 2001 Indian Census (Huenerfauth, 2002). These statistics and others clearly convey the global scale of this problem.

Making the Case for Including Functional Illiteracy

The lack of research addressing literacy and learning disability in the HCI discipline was understandable 10 or 15 years ago when this population was highly unlikely to interact with technology outside of an educational setting. With the ubiquitous nature of information technology in modern society (for example, the Internet, computers in all aspects of work, public kiosks, ATMs, electronic voting machines, e-health, and consumer electronics) this research gap is now indefensible. Accessible technology is no longer a luxury or an option; rather it is a prerequisite to gaining access to information that affects one's quality of life, participating in government, and contributing to the economy. A report from the Department of Commerce (2002) stated that "[b]etween December 1998 and September 2001, Internet use by individuals in the lowest-income households (lower socio-economic groups represent a disproportionately high percentage of the lower literacy population) increased at a 25 percent annual growth rate" (p. 56). The study also reported that "as of September 2001, about 65 million of the 115 million adults who were employed and 25 or over use a computer at work" (p. 57). While the percentage of workers using computers was significantly higher for "professional" positions, 20% of those working as operators, fabricators, laborers, farmers, and fishermen use a computer as part of their work (p. 58). Lastly, shifts in the economy from "hands-on work" to "information-and-technology work" will seriously disadvantage lower-literacy groups unless there is a corresponding increase in functional literacy skills.

If the size of this population alone were not enough to mobilize action, the cognitively disabled are a protected group under the Americans with Disabilities Act. Under the act, employers are required to provide workplace accommodations for employees who disclose their disability. That last stipulation of the act could very well be one of the reasons this population has been marginalized. Low literacy carries a stigma for most people and creates a reluctance to disclose—no disclosure, no requirement for accommodation.

Functional illiteracy and healthcare. Perhaps the most poignant case of a digital divide between the literate haves and the functionally illiterate have-nots is in the healthcare community. After a long delay, this community has made tremendous progress in implementing information technologies to enhance the quality of patient care, particularly in disseminating health-related information, managing patient care, accessing patient information, and promoting health services. As a result, the trend for seeking health-related information online is increasing dramatically. An estimated 70 million Americans have sought health information online (Cain, Mittman, Sarasohn-Kahn, & Wayne, 2000) from the nearly 10,000 or more health-related websites (Benton Foundation, 1999). Birru et al. (2004) estimated that between 40% and 54% of patients use the Internet to search for information on ailments and treatments. This development would be exciting, given the documented contributions of information to improved health, if it were not for the fact that equal access to this information does not exist for all Americans. Similar to the progression that played out in the larger development community, initial design efforts in the healthcare sector have focused first on the "typical" fully functioning user and then on the aging and physically disabled. Most healthcare materials—web-based or paper—are written well above the literacy level of the average American, approximately a 10th-grade level or greater. (Doak, Doak, & Root, 1996; Birru et al., 2004; Weiner et al., 2004).

The healthcare sector has both a moral and economic responsibility to address the needs of lower literacy populations. Morally, it is only right that all citizens benefit equally from the value information technology brings to the quality of healthcare services. Cashen, Dykes, and Gerber (2004) provided support for this position by reporting that literacy is a better predictor of health status than age, income, employment status, education, or race. There is clearly an economic motivation as well. A conservative estimate places excess healthcare costs tied to low literacy at $73 billion a year. (Rudd, Moeykens, & Colton, 1999). It is not surprising, then, that low-literacy adults are twice as likely to be hospitalized as their functionally literate counterparts (Birru et al., 2004).

In recent years the definition of the digital divide has shifted from economics to functional literacy. The availability of low-cost equipment and easy access to the Internet in public facilities has lessened the economic barriers. According to Shneiderman (2000), poor interface and interaction design remain as one of the defining variables between the technology haves and have-nots. And while we might hold out hope that we may one day eradicate illiteracy at the source, the fact that learning disabilities are lifelong and that fewer than one in eight low-literacy workers receives literacy training in the workplace (Sum, 1999, p. 156) strongly suggests that the burden falls on better design and appropriate accommodations.

CHARACTERISTICS OF THE POPULATION

Functional illiteracy is often referred to as the "invisible" disability. In many cases, the individual is unaware of the disability or the individual "hides" the disability. Findings from The National Literacy Survey (1993) suggest that as many as "66 to 75 percent of the adults in the lowest [literacy] level and 93 to 97 percent in the second lowest level described themselves as being able to read or write English 'well' or 'very well'" (The Literacy Skills of American Adults). In another study by Moon, Cheng, Patel, and Scheidt (1998), 70% of the participants reported they read "really well," while in actuality their reading scores reflected a 7th- to 8th-grade ability. Also contributing to the "invisibility" of this disability is the stigma attached to functional illiteracy and learning disabilities that increases an individual's reluctance to disclose the problem.

The learning disability label encompasses a wide range of information processing disorders, such as dyslexia (language), dyscalculia (mathematics), and dysgraphia (handwriting). They are thought to be neurobiological in origin and present themselves in various combinations and levels of severity. In addition, an individual with a deficit in one area may have strengths in other areas. Understanding the heterogeneous nature of the learning-disabled population is critical to formulating appropriate accommodations. Learning disabilities generally persist over a lifetime, but the manner in which the disability presents itself will change with life stages and accommodation strategies adopted by an individual (Gerber, 1998). The lifelong persistence of the disability, combined with the aforementioned lack of adult-literacy training, suggests this problem will not go away.

Overall, learning disabilities affect an individual's ability to develop and use reading, writing, reasoning, and mathematical skills (Karande, Sawant, Kulkarni, Galvankar, & Sholapurwala, 2005). In the absence of these fully developed skills, the individual's ability to access, process, and retain (learn) information is severely constrained. In addition, the many challenges and failures experienced by the learning disabled over a lifetime often result in a poor concept of self-worth and low self-esteem (Gerber, 1998).

One point at which some in the learning-disabled population break with the larger functionally illiterate community is at the level of underlying intelligence. Many individuals with one or more learning disabilities have normal or above normal intelligence (Rowland, 2004; Doak et al., 1996; Gerber, 1998). Consequently, once appropriate accommodations are provided to mitigate the effects of the disability, adequate intelligence exists to accomplish most cognitive tasks.

Functional illiteracy and aging. One remaining characteristic of this population is the disproportionately high rates of lower literacy levels in the aging population. Kirsch et al. (1993) reported, "Older adults were more likely than middle-aged and younger adults to demonstrate limited literacy skills. For example, adults over the age of 65 have average literacy scores that range from 56 to 61 points (or more than one level) below those of adults 40–54 years of age" (p. 5) (The Literacy Skills of America's Adults). In the most recent national literacy survey, 14% of the adult population is below basic prose literacy skills, with 26% of this group 65 years of age or older (NAAL, 2003). The cause for the disproportionate representation of older adults is likely a combination of a number of factors including higher school dropout rates for the older population, lack of special education services when they were in school, and the general decline in cognitive abilities associated with aging. Regardless of the cause, design accommodations that benefit lower-literacy populations are likely to have the added benefit of supporting the aging population as well.

The next section will move from this general overview to a detailed discussion of the functional characteristics of the learning disabled. A sizable portion of the research that fuels this discussion is from the study of dyslexia. The rationale for this focus is twofold: (a) dyslexia is the most common of the learning disabilities, affecting nearly 80% of the learning-disabled population (Karande et al., 2005); and (b) deficits associated with dyslexia are most likely to impact information processing as it relates to HCI.

Functional Characteristics of the Learning Disabled

Because of the tremendous diversity in the learning disability community and the high degree of overlap among discrete disabilities, it is easy to become overwhelmed by the complexity of the area. A more manageable strategy is to identify the functional characteristics shared across disabilities, with close attention to those that affect a particular interaction environment (Brown & Lawton, 2001; Bohman & Anderson, 2005). Figure 44.1 summarizes a number of proposed groupings.

After carefully examining these characteristics, I simplified this list into four dynamically interconnected categories:

1. Reading
2. Memory
3. Metacognitive
4. Search and navigation

These categories were selected because of their likely affect on the computer-interaction environment. In addition, many of the characteristics in Table 44.1 are directly linked to deficits in a common underlying process, such as metacognition or working memory. These four categories, as highlighted in the discussion that follows, are not mutually exclusive. For example,

Source	Deficiency
Rowland (2005)	• Perception and processing • Problem solving • Low resilience to frustration—not persistent • Memory • Attention
Jiwnani (2001)	• Difficulty with sequential operations • Difficulty with complex, cluttered displays and controls • Difficulty choosing from large sets • Timed responses— particularly when they involve text
Bohman and Anderson (2005)	• Memory • Problem-solving • Attention • Reading, linguistic, and verbal comprehension • Math comprehension • Visual comprehension
Cromley (2005)	• Poor decoding • Limited background knowledge • Low vocabulary • Dysfunctional beliefs about reading • Low-strategy use • Motivational Barriers • Working memory issues
Kolatch (2000)	• Trouble with abstract reasoning and organization • Short and long-term memory loss • Word finding and syntax development difficulties • Diminished capacity to organize, assimilate, and retain information (web section) (Introduction)
Web Aim	• Difficulty interpreting what is heard and read • Difficulty connecting information
The International Dyslexia Association	• Difficulties with accurate and/or fluent word recognition • Poor spelling and decoding abilities • Deficit in the phonological component of language • Problems in reading comprehension and reduced reading experience that can impede growth of vocabulary and background knowledge
The TRACE Center (2003)	• Memory difficulty recognizing and retrieving information • Perception difficulty taking in, attending to, and discriminating sensory information • Difficulties in problem solving • Difficulty evaluating outcomes • Difficulty generalizing previously learned information

FIG. 44.1. Functional characteristics of the learning disabled.

skilled reading draws on memory and requires the deployment of metacognitive strategies. Similarly, search and navigation place a load on working memory, draw on long-term memory, and require metacognitive monitoring. The following section will review each of these categories more closely, and the final section will offer recommendations for accommodating each of these categories in system design.

Reading

Given the prominent role of language in everyday communication as well as the more demanding software and web environments, reading is one of the first barriers encountered by the functionally illiterate. The alarming rate of poor prose and document-literacy skill was noted earlier. Also noted was the fact that poor reading is often tied to an underlying learning disability, inadequate schooling, or lack of exposure to reading. As in so many other cases in this area, an individual can suffer from all three conditions, two of them, or just one.

In dyslexic children, the learning disability affects reading at the most basic levels of phonological processing (Snowling, Deftry and Goulandris, 1996). Deficiencies at this basic level lead to further difficulties acquiring the complex skill of reading, building a rich vocabulary, and decreasing word-retrieval times. Most significantly, disabled readers allocate a disproportionately high amount of their attention to decoding letters and words, a process that is quickly automated by the nondisabled reader (van Gelderen et al., 2004). For the skilled reader, the process of automation frees the attention resource to focus on comprehension, the most critical component of the reading act. Skilled reading requires the management of a complex series of parallel actions, such as recognizing words, connecting those words to what came before, using those words to anticipate what will come next, and relating this combined experience to what one already knows. For the learning disabled, focusing most of the available attention on recognizing letters and decoding words severely compromises comprehension. Birru et al. (2004) reported that even when poor readers were able to read a passage, their inability to express answers related to the passage in their own words suggested minimal comprehension. Naturally, because reading is such a frustrating experience for the learning disabled, they are also less likely to read on a frequent basis, lowering the likelihood they will increase their reading skill through practice (Stanovich & West, 1989).

Given the difficulty experienced by this population at the most basic level of decoding, it becomes clear that the problem is greatly exacerbated by materials written at a grade level beyond their ability. When information is presented at a level beyond the capability of the reader, mental workload is increased significantly and comprehension is greatly diminished. From an accessibility perspective, grade-level readability is a persistent problem throughout the web development community. A recent study by West (2003) found that:

> "89% of government websites are not easily accessible to the citizenry because the site read at higher than an eighth grade level of literacy. Fully two-thirds of all sites have language consistent with a 12th grade reading level, which is higher than the average American [half of all Americans read no higher than the eighth grade level]." (p. 3)

In another review conducted by Croft and Peterson (2002) the investigators found the mean readability score of 145 asthma-related websites was above the 10th grade, with 27 of the websites at the maximum 12th-grade level. Finally, Graber, D'Alessandro, and Johnson-West (2002) conducted a study of the readability level of privacy policies on Internet health websites. The investigators found that the average readability of the statement was

at a level equal to that of a second-year college student, far outside the reach of the average American.

Despite the presence of a reading disability, most of this population possesses average or above average intelligence. As a result, if materials are presented at the appropriate grade level, the learning-disabled reader is capable of understanding. In separate studies conducted by Nielsen (2005) and Birru et al. (2004), poor readers performed well when interacting with easier-to-read materials.

While there remains considerable disagreement on how best to measure grade-level reading, certain measures and tests have gained wide acceptance. One such test, the Flesch Kincaid Grade Level Readability test, measures readability by dividing the sentence length (number words divided by the number of sentences) by the average number of syllables per word (number of syllables divided by the number of words). The resulting calculation is then correlated with an appropriate grade level. While this is a far from perfect measure, it does provide a useful benchmark of readability when evaluating the suitability of text for a given population. Unfortunately, this test along with others, such as the SMOG Readability Formula and the FRY Graph Reading Index, apply only to prose and not to lists, labels, and headings. With this type of text, the use of familiar words improves accessibility. Designers can consult reference works, such as the *Word Frequency Book,* (Carroll, 1972), to identify the most commonly used words in the English language. A more complex form of language support can be found on the CAST website (http://www.cast.org/) where the sponsors provide a language tool to support low-literacy users visiting the CAST site.

Because the functionally illiterate commonly experience difficulty with reading, illustrations, audio output, and video are often offered as effective accommodations. While illustrations can assist the less-able reader (Doak et al., 1996; Weiner et al., 2004), positive effects vary with the reader and nature of the information. According to Beveridge and Griffiths (1987), illustrations contributed positively to reading performance in easier-to-read passages. In contrast, they also noted that illustrations degraded reading performance in the most difficult passages. In a study related to the supplemental use of animation, Larsen (1995) reported learning-disabled children had trouble due to the distraction of the animation. Similarly, Jiwnani (2001) reported system designers must be careful using auditory output since it can confuse the learning disabled. The pace of the output must be slow, free of background noise, and repeatable under the control of the user.

Clearly, interface and interaction designers must choose word, sentence length, and sentence structure with due consideration for the readability level of the audience. Readability can be seen as the first barrier in software or web design encountered by the functionally illiterate population. If this population is unable to process the underlying language used in the display, all other accommodations are meaningless.

Memory

Long-term memory. The long-term and working memory systems each have implications for the functionally illiterate and learning-disabled population. As noted in the previous section, this population is less likely to read as often and, consequently, fails to benefit from a major source of acquiring domain knowledge. Without well-established domain models, interacting with web-based information or a software system is considerably more demanding. Cognitive science has long recognized that long-term memories supplement and extend the limited capacity of working memory. This topic will be explored further in the navigation-and-search section that follows.

Working memory. Cognitive science has also recognized the limited capacity of working memory and the major bottleneck it represents in information processing and human-computer interaction. While learning disabilities are very diverse in their causes and in how they manifest themselves, one consistent variable in all learning disabilities is working memory capacity. The dynamic interaction between memory capacity and the time information can be actively maintained also imposes severe constraints on the learning disabled. Debate has raged for years whether the learning disabled suffer from diminished working memory capacity or whether the complex dynamics of the underlying cognitive deficiencies place an excessive burden on a "normal" capacity (Daneman & Carpenter, 1980; Swanson, 1993; Vellutino et al., 1996; Ransby & Swanson, 2003). From the perspective of design accommodations, the precise source of the problem is insignificant.

Clearly, the attention devoted to low-level decoding places a heavy demand on working memory capacity. In addition, the increased likelihood of weaker mental models minimizes the opportunity to "offload" processing burden from the working memory. Further, the load imposed by metacognition, search, and navigation—as discussed in the sections that follow—places additional demands on this precious resource. Finally, the high level of anxiety and frustration experienced by this population also operates in working memory and will negatively affect performance on an information-processing task (Lee, 1999). Consequently, this issue, more than any other, guides many of the design and support recommendations in the final section. For example, limited working memory resource accounts for the problems the learning disabled experience with sequenced operations in software or in retracing their paths of travel navigating a website. Interface and interaction designs must compensate for the limitations of working memory or suffer the consequences of lower performance, increased errors, or abandonment of the task.

While the underlying learning disability can make the acquisition of expertise more difficult, it would be wrong to think this population does not bring some level of learning and a variety of conceptual models to an interaction experience. As with so many other variables in accessibility design, we must recognize that the goals and models of the lower-literacy population may not align with those of the general population. Dickinson, Eisma, and Gregor (2003) reported that "users made remarks which indicated that they were not trying to understand the system or find generic rules that could help them to use it better" (p. 63). In other words, they were not attempting to understand the model underlying the system; they were simply trying to get something accomplished. System design should focus on that goal and avoid excessive functionality and extraneous information. From a development perspective, it is always easy to imag-

ine occasions when one user or another might need this feature or that piece of information. Unfortunately, the lower-literacy population comes to the system with a greatly limited need, for both functions and information. Accessible design must consider these needs. Interface clutter increases mental workload on users as they attempt to locate information and features that align with their needs, a difficult challenge given their attentional impairments.

Metacognitive

Another critical deficiency in most dyslexics is in metacognition, the process of thinking about thinking. Metacognition can take many forms, including strategic planning, monitoring, self-appraisal, document-processing strategies, and reading strategies. Metacognition operates actively in working memory, parallel to the main information-processing task. Consequently, this activity places yet another demand on the memory resource. Also, because these strategies are learned behaviors, it suggests an interaction with long-term memory. Research has shown that the learning disabled are generally deficient in the metacognitive area. While nondisabled learners acquire most of these strategies without formal instruction, studies have shown these strategies can effectively be taught to the learning disabled (Collins, Dickson, simmons, & Kameenui, 1998). Again, we must be reminded that skilled readers become skilled through years of practice with different reading materials that were processed to support diverse purposes. Because proficient readers and learners use these strategies on a frequent basis, they migrate, over time, to the level of an automatic skill. In contrast, the absence of persistent exposure to reading places the learning disabled at a further disadvantage.

Cromley (2005) indicated that the learning disabled are less likely to know when they do not understand and are less likely to reread, synthesize, generate questions, or make predictions. They are also more likely to "satisfice" and accept partial or incorrect information rather than persevering to gain a more complete understanding. A distinguishing attribute of good readers is their ability to monitor the information-processing activity; in other words, they ask themselves: "Do I understand what I am reading? Do I see the relevance of the information to the task? What do I expect to follow in the next passage? How does this relate to what I already know?" Corley and Taymans (2002) organized this activity in three parts: setting goals before the task and establishing a plan; monitoring comprehension and understanding while engaging the task; and evaluating one's learning after the task and making adaptations when faced with similar tasks. Danielson (2002), referencing users navigating the web, suggested that once goals are made, strategies are selected for achieving that goal. Further, strategies are constantly assessed and adjusted many times in a successful interaction experience. Given the mental demand of these activities, monitoring and adjusting activities pose a challenge for the disabled. Finally, Britt and Gabrys (2002) indicated that document literacy requires four metacognitive skills: sourcing (evaluating the credibility of source), corroboration (seeking independent confirmation), integration (creating mental representation of information), and search (a skill woven throughout the other three areas). The inability of the learning disabled to engage in self-evaluating activities is evident in the work of Birru et al. (2004). Although most of the participants in their study failed to answer the questions correctly, seven out of eight reported feeling "very comfortable" or "comfortable" with their Internet search experience. On the sourcing and corroboration skills, five out of eight used information provided on sponsored sites, yet still reported it was "very easy" to find trustworthy information on the Internet.

In short, metacognition requires a tremendous amount of cognitive activity, activity that occurs in parallel with the primary decoding task. All of this activity occurs in the executive control component of working memory. As noted in the preceding section, this is a limited resource for the learning-disabled reader who devotes much of this resource to low-level decoding.

We take for granted that nondisabled users engage in all of these activities effectively and effortlessly. Design support for this variable amounts to providing constant feedback to the disabled user, explicitly connecting the information to the task, establishing checkpoints where the system asks questions of the reader to monitor understanding, and engineering performance support in the form of cognitive scaffolding. On the matter of feedback, for example, Dickinson et al. (2003) noted that longer tasks are tolerated by the learning disabled so long as they receive frequent feedback on what is happening and confirmation they are proceeding correctly. In the absence of such feedback, the poor reader becomes anxious, a state that further degrades performance since anxiety occupies the working memory space.

Navigation and Search

The learning disabled have traditionally experienced tremendous difficulty reading paper-based materials. They experience this difficulty despite the fact the linear format found in most paper-based materials offers considerable support for the reader. In these materials, the author assumes the burden for communicating organization through meaningful structures, logical sequences, helpful transitions, and explicit connections. In contrast, the nonlinear web requires the reader to infer organization and build a coherent model of the subject matter as the information unfolds in less predictable ways (Britt & Gabrys, 2002). McDonald and Stevenson (1998) identified disorientation in the nonlinear environment as one of the most challenging elements of the web. While disorientation is an obstacle that can be overcome by able users, this barrier is often insurmountable for the learning disabled. This population, at best, experiences difficulty building a mental model of the information, monitoring where they are in the experience, or retracing their path. Each of these activities draws on their overtaxed working memory.

As one examines the factors affecting readers' ability to navigate, a number of variables emerge:

- Basic navigation skills related to the interface
- Topology formats: nonlinear, hierarchical, and mixed topologies
- Depth versus breadth of the structure
- Navigational aids

Interaction skills. While there is limited research examining the effects of learning disabilities on navigation, the work

of Zarcadoolas, Blanco, Boyer, and Pleasant (2002) suggested learning-disabled users can quickly learn and retain basic web-interaction skills. These skills include the behavior of links and the act of scrolling. Participants in this study did require training and prompting to use graphic links. Overall, the positive reaction to the linking convention is encouraging since Zarcadoolas et al. also found the participants, when given a choice, preferred following links rather than using search. Worth noting, Birru et al. (2004) found that users in their study, although comfortable with the linking convention, would seldom click on more than one or two links to answer questions. Finally, although basic scrolling behavior was understood, none of the participants in the Zarcadoolas et al. study scrolled to view additional information. Without additional corroborating evidence it is difficult to generalize from these limited studies. However, the behaviors exhibited in these studies warrant close scrutiny as we continue to study and observe this class of users.

In contrast to the relative ease with which this population learned basic interaction behaviors, a more significant challenge is faced navigating the larger system. Danielson (2002) suggested users begin the navigation task with a decision about whether they are looking for a specific item, a group of items, or general information about the contents of a domain area. As noted in the metacognitive area, the learning disabled are generally weak at planning tasks, which immediately places them at a disadvantage in the navigation area. Unfortunately, there is little or no research that directly examines the learning-disabled population and the structural issues of nonlinear versus hierarchical, or depth versus breadth. There is, however, considerable speculation on these issues since performance differences for general-population users are typically discussed within the context of working memory capacity.

Topology. The work of McDonald and Stevenson (1998) shed some light on the issue of the efficacy of hierarchical, nonlinear, and mixed (hierarchical with referential links) topologies. Most significant perhaps for the purposes of this discussion, they found novices benefited most from the mixed structure since it offered a balance of freedom and control. The referential links supported exploration of a site without the support of a well-formed mental model, while the hierarchical framework served to constrain movements and minimize disorientation. While the strict hierarchical structure provided the greatest control and guidance for participants, it proved inefficient when participants wanted to make distal movements in the structure. Finally, the nonlinear structure simply provided too many options and placed the heaviest demands on expertise and working memory. In an earlier study, McDonald and Stevenson (1996) found users stopped reading too soon when faced with too many decisions related to "what" and "how much" to read. In each of these situations, it is easy to speculate that the problems faced by the participants in these studies would only be exacerbated for the learning disabled, given their underlying deficiencies in working memory and metacognition.

Depth vs. breadth. The efficacy of breadth versus depth in the hierarchy is also open to speculation. As with topology, the depth issue is framed by limitations of working memory and the users' ability to distribute the navigation workload to their own conceptual models of the domain. In this discussion, we see greater diversity of opinion. In the world of web design, best practice favors breadth over depth, resulting in a greater number of choices at the highest level. A study by Larson and Czerwinski (1998) suggested that a medium depth produced the best search performance over either the broader or the deeper options. Benard (n.d.) suggested depth versus breadth is not the best framing mechanism. Instead, he suggests the shape of the hierarchy is the best predictor of search performance, finding a broad first level, a narrow middle level, and a broad base to be most efficient. While not based on a controlled study, Kolatch (2000) proposed a simpler top-level interface, offering fewer choices and a deeper structure for cognitively disabled users. Kolatch was careful to point out that this contradicts most research on the topic (web section Analyses). Until new research suggests otherwise, the balanced structure proposed by Larson and Czerwinski (1998) provides guidance consistent with our understanding of the adverse effects of too many choices and the mental load imposed by deep structures.

Navigation aids. Danielson (2002) suggested that a variety of visual navigational aids can mitigate the aforementioned problems encountered in navigation. Navigational aids fall into two categories: index and table of contents lists, and site maps. Danielson noted that graphical site maps are superior to lists and table of contents at conveying the relationships between distal nodes on the site. The study also showed that the site maps benefited nonknowledgeable users more than the knowledgeable. Finally, when compared to a control group without site maps, Danielson reported that subjects with maps were less likely to abandon the task, reported information-seeking confidence, moved deeper into the site, and made more movements outside the hierarchy. Given the previously defined tendencies of the learning-disabled population, it's highly probable that they would benefit from the assistance offered by site maps. Supporting this theory, Mirchandani (2003) found detailed site maps helped the disabled user find information within a click or two. See the ADA Insights website (http://www.adainsights.org) for an example of a site map designed to support disabled users. As a visualization, the benefit of the site map is its ability to shift the cognitive load from working memory to the visual and thereby transform the load-intensive cognitive task to a less labor-intensive perceptual task. A more extensive navigational support tool can be found on the CAST website (http://www.cast.org/) The tool displays a taxonomical view of the site, the user's current location, the user's navigational path, quick links to frequently visited areas, and a notepad that allows the creation and storage of notes. Combined, this support helps compensate for deficiencies in monitoring skills.

Search. In contrast to navigation, research is slowly emerging related to the learning disabled and the search task. Not surprisingly, this research reports the learning disabled experience tremendous difficulty with search tasks. At the most basic level, the poor spelling displayed by many in the lower-literacy population creates a major barrier to successful searches. Search engines, such as Google, that offer alternative spellings are best. Weak self-monitoring skills also decrease the likelihood that the user will recognize that a mistake was made. Even when search

terms were spelled correctly, Birru et al. (2004) found that subjects rarely retyped search terms to locate more relevant items—a serious problem since they also found the low-literacy adults in this study rarely used optimal search terms as they attempted to retrieve health-related information. Birru et al. (2004) concluded by highlighting the benefits of a categorizing search engine as one means of minimizing the adverse effects of poor search and long lists of results. This engine would sort results by category and minimize the need to evaluate long, unsorted lists. This is a good example of a technology assuming the burden for a weak metacognitive skill in the disabled population.

DESIGN ACCOMMODATIONS

There are three approaches to accommodating the needs of the learning-disabled population. Each of these recommendations is based on the previously discussed characteristics of this population.

1. Assistive technologies: Building technologies customized to compensate for the disability;
2. Layered design: Designating special sections of the website or software for a disabled population;
3. Universal design: Adopting design practices that enable the learning-disabled population while benefiting the larger population as well.

The following sections will briefly review the first two categories and provide more detailed recommendations for the third category, universal design. This emphasis is based on three considerations. First, the initial two categories require greater effort and increase development costs. Second, they benefit a smaller portion of the population, typically the most severely disabled. Third, universal design principles are less likely to increase development time or costs and will likely benefit a larger number of the disabled and fully functioning populations.

Assistive technologies. A number of studies have demonstrated the benefits of assistive technologies for the disabled (Brown, 1992; Cole & Dehdashti, 1998; Newell & Gregor, 2000; McGrenere et al., 2003). A typical application in this genre is a stand-alone computer-based educational system, designed to assist severely disabled children. Other, less intrusive accommodations, including software agents and embedded scaffolding, have been explored by Shaw, Johnson, and Ganeshan (1999); Quintana, Krajcit, and Soloway (2003); and Shneiderman (2003).

Scaffolding and agents offer performance support to the disabled user in the early stages of interaction with a new product. Shneiderman (2000) described this approach as "evolutionary learning." As the user becomes more proficient with the requirements of the system and task, the scaffolding is slowly torn down, either under the control of the user or through intelligent software agents. Most typically, the agent replaces or supports a deficient metacognitive process by assisting with planning, monitoring, self-appraisal, strategy setting, and feedback. Quite simply, this burden is shifted from the users' working memory to the system. Shneiderman (2000) and Quintana, Krajcik, and Soloway (2002) highlighted the efficacy of this approach for supporting less-knowledgeable and disabled populations.

Layered design. The layered design strategy requires the design of two versions of the system: one for the functionally literate users and one for the functionally illiterate. While it is generally preferred not to separate disabled and fully functioning populations, layered design is simply an extension of a practice that has been used for years for separating domain experts from nonexperts or native speakers from non-native speakers. In this case, one area of the site would be written at the population average 8th-grade level, while another section of the site would be written at the 5th-grade level or lower. Lobach, Arbanas, Mishra, Campbell, and Wildemuth, (2004) described the benefits of a two-tier literacy system in their study. Sites such as this may include other assistance, such as agents, site maps, illustrations, and the like.

Universal design. In contrast to the first two approaches, universal design accommodations are seamlessly embedded in the system interface and interaction design. This class of accommodations is less likely to affect development costs or create an obstacle for the nondisabled. Consequently, these accommodations are more likely to be embraced by the wider development and user communities. Further, it is widely accepted in the accessibility community that these accommodations improve the usability of the product for all users (Shneiderman, 2000; Dickinson, Eisma, & Gregor, 2003). The following recommendations are organized in the previously defined categories, although support for a given recommendation is often found in multiple categories. Finally, recommendations designated with an identify recommendations that require additional research and warrant close monitoring. All others enjoy wide support in the literature.

Reading

When designing the language component of any system, support the following:

- Maintain an appropriate reading level (use readability formulas to establish a benchmark; measure word difficulty and sentence length);
- Use active voice;
- Use the *Word Frequency Book* to identify common words;
- Place information and instruction in context;
- Employ lists;
- Chunk information;
- Present content in sequence;
- Repeat information from screen to screen (do not assume the user will carry over);
- Maintain consistency in language and procedures;
- Communicate directly and concretely;
- Emphasize actions users must complete;
- Highlight critical information, information structure, or new information;
- Use familiar terms, and avoid acronyms and jargon;

- Use visual and auditory prompts;
- Provide definitions of critical terms through direct linking to glossary;
- Use illustrations to complement text, communicate structure, and emphasize connections;
- Highlight (using circles, arrows, and the like) critical areas of an illustration and explicitly connect to the text;
- Avoid the gratuitous use of animations and other movement;*
- Use 12-point type;
- Use familiar typefaces (there is conflicting research regarding the efficacy of serif versus sans serif);*
- Avoid tight letter-spacing (makes low-level decoding more difficult);
- Pace auditory output slowly, and allow user control to repeat output;
- Avoid background noise with auditory output;
- Provide "looser" versus tighter line spacing;
- Support easy, user controlled, style changes;
- Make it "look easy" by lowering information density;
- Limit use of italics and uppercase (effects are much more severe than for the general population);
- Maintain higher contrast;*
- Avoid light text on a dark background;
- Use ragged right formats to preserve consistent word space.

Memory

In an effort to decrease mental workload, support the following:

- Maintain consistency in all aspects of the design;
- Leverage existing knowledge, behaviors, and tasks;
- Avoid splitting attention between two tasks;
- Focus on the user goals;
- Limit information and features to what is really needed;
- Limit chunking complexity for audio output (particularly IVR);
- Focus on behaviors and tasks rather than facts;
- Partition tasks in reasonably sized groups;
- Avoid the need to retain information over long tasks or across multiple screens;
- Support mental calculations, decisions, and comparisons;
- Complete mathematical calculations;
- Limit choices;
- Provide a list of options for entry fields;
- Complete information automatically in forms and fields whenever possible;
- Employ advanced organizers;
- Use mnemonics;
- Minimize screen clutter;
- Provide extra time for tasks;
- Eliminate the anxiety of timeouts.

Metacognitive

In an effort to strengthen or replace deficient metacognitive processes, support the following:

- Maintain a consistent design;
- Convey associations between new information or process and that which is known;
- Communicate goal or purpose of the site immediately;
- Communicate prerequisite knowledge for the task and provide convenient links;
- Communicate required sequences or organizational structures;
- Allow the user to interact with the information;
- Use checklists to support self-monitoring;
- Minimize choices that require fine discriminations or close monitoring;
- Design for immediate/early success (which lowers anxiety and builds confidence);
- Align with user goals, and provide reward or convey value proposition (motivation);
- Support cooperative work activities;
- Provide reminders;
- Provide source information for material presented;
- Minimize embedded links;
- Avoid taking user to other pages for ancillary information;
- Employ error prevention and recovery support;
- Query when choices or decisions are required;
- Review information entered or validate the successful completion of a process;
- Use auditory and visual cues to mark stages of a work cycle, helping the user self-monitor.

Navigation and Search

In an effort to improve the effectiveness of search and navigation, support the following:

- Make information and features supporting goals readily accessible to minimize navigation and search;
- Provide persistent presentation of path history;
- Offer persistent opportunity to exit, backup, or return to start;
- Use redundancy: repetition, signal design, and channel;
- Provide status indicators;
- Minimize scrolling;
- Label all links;
- Provide linked paths for probable scenarios;
- Maintain alerts onscreen until dismissed by user;
- Place information or process in context;
- Partition information into categories defined by clear rules;
- Use site maps, tables of contents, and indexes;
- Use a topology that is primarily hierarchical with referential links to areas aligned with goals;*

- Maintain a medium breadth and depth—not too broad, not too deep;*
- Use clear, thematic labels at each level;
- Provide productive terms for search;
- Offer suggested spellings;
- Use categorizing search engines;
- Provide performance support for evaluating search results (corroboration and sourcing).

Guidance for Usability Testing

To produce the best data and protect the well-being of participants, support the following:

- Avoid embarrassment or humiliation (anxiety lowers performance);
- Consider ethical obligations when screening for lower-literacy samples (full-disclosure issues, informed consent, and IRB);
- Minimize anxiety in testing situation;
- Test in context (actual work environment);
- Avoid using the term "testing" since this population equates the term with failure;
- Break tasks in the test script into small, yet logical, units;
- Shorten test times to minimize effects of fatigue on performance and to compensate for attention deficit;
- Use direct interaction protocol since users may not freely think aloud (inadequate working memory to support both think-aloud and to acclimate to the task);*
- Accommodate slower processing speeds when setting performance levels (e.g., task times);
- Expect participants to report exaggerated levels of performance or system quality in self-reporting as a means of covering poor literacy skills.

CONCLUSION

Ten years ago, our concern for the effects of lower literacy was confined exclusively to the processing of print media. With the explosion of information technology in the workplace and the role of the Internet as the repository for information of all types, literacy has become a major barrier for millions of Americans as they compete for work and attempt to improve the quality of their lives. While the problems encountered by lower-literacy populations with print media are significant, the problems become greatly exacerbated with the increased mental workload imposed by computer systems.

As advocated in this chapter, many of the barriers to accessibility for this population require simple modifications in the interface and interaction design. Accessible design is an art, in which the needs of one group are carefully balanced against those of another, and the designer recognizes that "accessible for most" is more achievable than "accessible for all." As one reviews the introductory list of design accommodations required by the lower-literacy population, HCI professionals will recognize most, if not all, of these variables as ones considered in each and every interface design. The difference—and a significant one—is that fully capable users have flexible learning, problem solving, and interaction skills that allow them to adjust to less-than-perfect design specifications. The learning disabled, in contrast, lack cognitive flexibility and suffer varying degrees of performance degradation in nonoptimum design environments. By increasing our understanding of this sizable population and embracing the design practices outlined here, we may finally ensure that universal accessibility is truly universal.

References

Benard, M. (n.d.). *Criteria for Optimal Web Design*. Retrieved January 13, 2006, from http://psychology.wichita.edu/optimalweb/structure.html

Benton Foundation. (1999). *Networking for Better Care: Health Care in the Information Age*. Retrieved December 28, 2003, from htpp://www.Benton.org/library/health/

Beveridge, M., & Griffiths, V (1987). The effects of pictures on the reading processes of less able readers: A mis-cue analysis approach. *Journal of Research in Reading, 10,* 29–42.

Birru, M., Monaco, V., Charles, L., Drew, H., Njie, V., Bierria, T., Detlefsen, E., & Steinman, R. (2004).Internet Usage by Low-Literacy Adults Seeking Health Information: An Observational Analysis. *Journal of Medical Internet Research, 6*(3), Retrieved December 12, 2005, from http://www.jmir.org/2004/3/e25/

Bohman, P., & Anderson, S. (2005). A Conceptual Framework for Accessibilty Tools to Benefit Users with Cognitive Disabilities. *Proccedings of the W4A at WWW2005*, ACM, 85–89.

Britt, M., & Gabrys, G. (2002). Implications of Document-Level Literacy Skills for Website Design. *Behavior Research Methods, Instruments, & Computers, 34*(2), 170–176.

Brown, C. (1992). Assistive Technology Computers and Persons with Disabilities *Communications of the ACM, 35*(5), 36–45.

Brown, D, &, Lawton, J. (2001). Design guidelines and issues for website design and use by people with a learning disability. Centre for Educational Technology Interoperability Standards. Retrieved March 28, 2007 from http://www.cetis.ac.uk/members/accessibility/links/disabilities/cogdis.

Cain, M. M., Mittman, R., Sarasohn-Kahn, J., & Wayne, J. (2000). *Health E-People: The Online Consumer Experience*. Oakland, CA: Institute for the Future, California Health Care Foundation.

Carroll, J. (1972). The American Heritage word frequency book. New York: Houghton Mifflin.

Cashen, M., Dykes, P., & Gerber, B. (2004). eHealth Technology and Internet Resources: Barriers for Vulnerable Populations. *The Journal of Cardiovascular Nursing, 19*(3), 209–214.

Cole, E., & Dehdashti, P. (1998). Computer-Based Cognitive Prosthetics: Assistive Technology for the Treatment of Cognitive Disabilities. *Proceedings of the Third International ACM Conference on Assistive Technologies,* ACM: Arrington, VA, 11–18.

Collins, V., Dickson, S., Simmons, D., & Kameenui, E. (1998). *Metacognition and its Relations to Reading Comprehension: A Synthesis of the Research*. National Center to Improve the Tools of Educators, University Washington.

Corley, M., & Taymans, J. (2002). *Adults with Learning Disabilities: A Review of the Literature*. Retrieved December 30, 2005, from http://www.ncsall.net/?id+575

Croft D. R., & Peterson M. W. (2002). An Evaluation of the Quality and Contents of Asthma Education on the World Wide Web. *Chest, 121*(4), 1301–1307.

Cromley, J. (2005). Metacognition. *Cognitive Strategy Instruction, and Reading in Adult Literacy*. Retrieved January 17, 2006, from http://www.ncsall.net/index.php?id=773

Daneman, M., & Carpenter, P. (1980). Individual Differences in Working Memory and Reading. *Journal of Verbal Learning and Verbal Behavior, 19*, 450–466.

Danielson, D. (2002). Web Navigation and the Behavioral Effects of Constantly Visible Site Maps. *Interacting With Computers, 14*(5), 601–618.

Dickinson, A., Eisma, R., & Gregor, P. (2003). Challenging Interfaces/Redesigning Users. *CUU 2003*. British Columbia, Canada, 61–68.

Doak, C., Doak, L., & Root, J. (1996). *Teaching Patients with Low Literacy Skills*. Philadelphia: J.B. Lippincott Company.

Dropout Rates in the United States. National Center for Educational Statistics. Retrieved December 10, 2003, from http://nces.ed.gov/pubs2002/dropoutpub_2001/11.

"Fact Sheet:" Workforce Literacy. National Institute for Literacy, Washington, DC, Retrieved January 4, 2006, from http://www.nifl.gov/nifl/facts/workforce.html

Gerber, P. (1998). Characteristics of Adults with Specific Learning Disabilities. Retrieved December 28, 2003, from http://www.idonline.org/ld_indepth/adult/characteristics.html

Graber, M. A., D'Alessandro, D. M., & Johnson-West (2002). Reading Level of Privacy Policies on Internet Websites. *The Journal of Family Practice, 51*(7), 642–645.

Gribbons, W. (1992). The Functionally Illiterate: Handicapped by Design. *Proceedings of the International Professional Communications Conference*. IEEE, 302–307.

Huenerfauth, M. (2002). Design Approaches for Developing User Interfaces Accessible to Illiterate Users. *American Association of Artificial Intelligence Conference (AAAI 2002), Intelligent and Situation-Aware Media and Presentations Workshop*. Edmonton, Alberta, Canada.

Jiwnani, K. (2001). *Designing for Users with Learning Disabilities*. Retrieved December 28, 2005, from http://www.otal.umd.edu/uupractice/cognition/

Karande, S., Sawant, S., Kulkarni, M., Galvankar, P., & Sholapurwala, R. (2005). Comparison of Cognitive Abilities Between Groups of Children with Specific learning Disability Having Average, Bright Normal and Superior Nonverbal Intelligence. *Indian Journal of Medical Sciences, 59*(3), 95–103.

Kickbusch, I. (2001). Health Literacy: Addressing the Health and Education Divide. *Health Promotion International, 16*(3), 289–297.

Kirsch, I. Jungeblut, A., Jenkins, L., & Kolstad, A. (1993). *Adult Illiteracy in America: a first look at the results of the National Literacy Survey*. Washington: Department of Education, Center for Educational Statistics. Washington, DC.

Kirsch,I, Jungeblut, A., Jenkins, L., & Kolstad, A. (1993). *Executive Summary of Adult Literacy in America: A first Look at the results of the National Literacy Survey*. Retrieved October 27, 2003, from http://www.nces.ed.govnaal/resources/execsumm.asp

Kolatch, E. (2000). *Designing for Users with Cognitive Disabilities*. Retrieved December 28, 2005, from http://www.otal.umd.edu/uuguide/erica/

Larsen, S. (1995). What is "Quality" in the use of Technology for Children with Learning Disabilities? *Learning Disability Quarterly, 18*(2), 118–130.

Larson, K. & Czerwinski, M. (1998). Web Page Design: Implications of Memory, Structure and Scent for Information Retrieval. *Proceedings of CHI* 98, USA, ACM, 25–32.

Lee, J. (1999, Spring). Test Anxiety and Working Memory. *The Journal of Experimental Education, 67*, 218–240. Retrieved December 4, 2005, from ProQuest database.

Lobach, D., Arbanas, J., Mishra, D., Campbell, M., & Wildemuth, B. (2004). Adapting the Human-Computer Interface for Reading Literacy and Computer Skill to Facilitate Collection of Information Directly from Patients. *MEDINFO 2004*, Amsterdam, IOS Press, 1143–1146.

McDonald, S., & Stevenson, R. (1996). Disorientation in Hypertext: The Effects of Three Text Structures on Navigation Performance. *Applied Ergonomics,* 27, 61–68.

McDonald, S. and Stevenson, R. (1998). Effects of Text Structure and Prior Knowledge of the Learner on Navigation in Hypertext. *Human Factors, 40*(1), 18–27.

McGrenere, J., Davies, D., Findlater, L., Graf, P., Klawe, M., Moffatt, K., Purves, B., & Yang, S. (2003). Insights from the Aphasia Project: Designing Technology For and With People who Have Aphasia. *CUU 2003*. ACM: Arlington, VA, 112–118.

Mirchandani, N. (2003). *Web Accessibility for People with Cognitive Disabilities: Universal Design Principles at Work*. Retrieved December 28, 2005, from http://www.ncddr.org/du/researchexchange/v08n03/8_access.html

Moon, R., Cheng, T., Patel, K., & Scheidt, P. (1998). Parental Literacy Level and Understanding of Medical Information. *Pediatrics, 102*(2). Retrieved from Medline.

National Assessment of Adult Literacy (NAAL). *A First Look at the Literacy of Americas Adults in the 21st Century*. (2003). National Center for Educational Statistics, U.S. Department of Education, Institute of Educational Sciences, NCES 2006-470, Washington, DC.

National Dyslexia Association (n.d.). *What is Dyslexia?* Retrieved November 10, 2003, from http://www.interdys.org

Newell, A., Carmichael, A., Gregor, P., & Alm, N. (2003). Information Technology for Cognitive Support. *In The Human Computer Interaction Handbook*. Eds. Julie Jacko and Andrew Sears. Mahwah, NJ: Lawrence Erlbaum Associates, 464–481.

Newell, A. & Gregor, P. (2000). User Sensitive Inclusive Design: In Search of a New Paradigm. *CUU 2000*. Arlington, VA: ACM, 39–44.

Nielsen, J. (2005). *Lower-Literacy Users*. Retrieved December 9, 2005, from http://www.useit.com/alertbox/20050314.html

Quintana, C., Krajcik, J., & Soloway, E. (2002). A Case Study to Distill Structural Scaffolding Guidelines for Scaffolded Software Environments. *CHI 2002*. Minneapolis, Minnesota: ACM, 81–88.

Ransby, M., & Swanson, H. L. (2003). Reading Comprehension Skills of Young Adults with Childhood Diagnoses of Dyslexia. *Journal of Learning Disabilities, 36*(6), 538–555.

Rowland, C. (2004). *Cognitive Disabilities Part 2: Conceptualizing Design Considerations*. Retrieved December 28, 2005, from http://www.webaim.org/techniques/articles/conceptualize/?templatetype=3

Rudd, R. E., Moeykens, B. A., & Colton, T. C. (1999). *Health and Literacy: A Review of Medical and Public Health Literature*. New York: Josey-Bass.

Shaw, E., Johnson, W., & Ganeshan, R. (1999). Pedagogical Agents on the Web. *Autonomous Agents 1999*. Seattle, WA, 283–290.

Sheiderman, B. (2000). Universal Usability. *Communications of the ACM, 43*(5), 84–91.

Shneiderman, B. (2003). Promoting Universal Usability with Multi-Layer Interface Design. *CUU 2003*. Vancouver, British Columbia, Canada, 1–8.

Snowling, M., Defty, N., & Goulandris, N. (1996). A Longitudinal Study of Reading Development in Dyslexic Children. *Journal of Educational Psychology, 88*(4), 653–669.

Stanovich, K., & West, R. (1989). Exposure to Print and Orthographic Processing. *Reading Research Quarterly, 24*, 402–433.

Sum, A. (1999). *Literacy in the Labor Force*. National Center for Education Statistics, US Department of Education, Washington, DC.

Swanson, H. L. (1993). Working Memory in Learning Disability Subgroups. *Journal of Experimental Child Psychology, 56*, 87–114.

TRACE Center, University of Wisconsin. *A Brief Introduction to Disabilities*. Retrieved November 23, 2003, from htpp://trace.wisc.edu/docs/population/populat.htm.

U.S. Department of Commerce. (2002). *A Nation Online: How Americans are Expanding Their Use of the Internet*. Economics and Statistics Administration, National Telecommunications and Information Administration. Washington, DC.

van Gelderen, A., Schoonen, R., de Glopper, K., Hulstijn, J., Simis, A., Snellings, P., & Stevenson, M. (2004). Linguistic Knowledge, Processing Speed, and Metacognitive Knowledge in First and Second Language Reading Comprehension: A Component Analysis. *Journal of Educational Psychology, 96*(1), 19–30.

Vellutino, F., Scanlon, D., Sipay, E., Small, S., Chen, R., Pratt, A., & Denckla, M. (1996). Cognitive profiles of Difficult to Remediate and Readily Remediated Poor Readers: Early Intervention as a Vehicle for Distinguishing Between Cognitive and Experimental Deficits as Basic Causes of Specific Reading Disability. *Journal of Educational Psychology, 88*(4), 601–638.

W3C (1999). Web Content Accessibility Guidelines 1.0 [Electronic edition] Retrieved April 3, 2007, from http://www.w3.org/TR/WAI-WebContent/

Weiner, J., Aquirre, A., Ravenell, K., Kovath, K., McDevit, L., Murphy, J., Asch, D., & Shea, J. (2004). Designing an Illustrated Patient Satisfaction Instrument for Low Literacy Populations. *The American Journal of Managed Care, 10*(11), 853–859.

West, D. (2003). *State and Federal E-Government in the United States*. Center for Public Policy, Brown University, Providence, RI. 1–29.

Zarcadoolas, C., Blanco, M., Boyer, J., & Pleasant, A. (2002). Unweaving the Web: An Exploratory Study of Low-Literate Adults' Navigation Skills on the World Wide Web. *Journal of Health Communication, 17*, 309–324.

•45•

COMPUTING TECHNOLOGIES FOR DEAF AND HARD OF HEARING USERS

Vicki L. H*anson*
IBM T. J. W*atson Research Center*

Introduction **886**
Hearing Loss **886**
Hearing and Language **886**
Display Technologies **887**
Audible Signals 887
Multimedia Interfaces 888
Captioning 888
Signing 888
Input Technologies **889**
Speech Interfaces 889
Sign Interfaces 890
Technology and Communication **890**
Conclusions **892**
Acknowledgments **892**
References **892**

INTRODUCTION

Traditionally, human interactions with computers has relied most heavily on visual perception and motor ability. Graphical user interfaces (GUIs) have dominated computer displays, both large and small, while mouse and keyboard devices have dominated computer input. Increasingly, however, newer interfaces are using sound and speech. The use of attention-grabbing multimedia computer presentations plus the increasing use of conversational speech interfaces are examples.

As exciting as these new technologies may be, they have the potential to disenfranchise deaf and hard of hearing users. This chapter will discuss interface technologies as they relate to the needs of users unable to hear auditory information. In addition, language considerations associated with limited hearing will be discussed.

This chapter begins with a discussion of hearing loss, followed by issues of language acquisition as they relate to hearing loss. Auditory user interfaces that present difficulties will be discussed next, along with information about interface alternatives that enable access for deaf and hard of hearing users. The chapter will conclude with a brief overview of technologies that have been developed to assist with communication.

HEARING LOSS

According to the National Institute on Deafness and Other Communication Disorders (NIDCD), about 28 million people in the United States have some degree of hearing loss (National Institutes of Health [NIH], 2004; see also, Mitchell, 2005, 2006). This is a sizable population to be considered in the design of computer interfaces. Numbers alone, however, obscure some significant differences among the individuals who experience hearing loss. Degree and type of loss, age of onset of loss, as well as family, educational, and societal influences will all contribute to the experience and abilities of an individual who has a functional hearing loss.

Some individuals will have relatively little hearing loss, while others will experience a profound loss. Individuals who are hard of hearing will generally have some hearing. The ability that an individual user will have to make use of their residual hearing, however, is not a straightforward calculation of dB loss. People with hearing loss will experience difficulty with pitch, timbre, and loudness, but, critically, will also experience difficulty with speech perception. Factors such as type of loss (e.g., conductive, sensorineural, mixed, or central) will have a major effect on the user experience. People with sensorineural hearing loss (such as resulting from lengthy exposure to loud noises or as a result of aging) generally will have more difficulty perceiving speech than people with conductive hearing losses that result from difficulties in middle ear functioning. The extent to which an individual makes use of this hearing for communication and whether the individual can hear computer sounds, however, varies greatly.

In addition, the way in which an individual having a hearing loss interacts with their environment may also be influenced by societal factors. Whether individuals identify themselves as Deaf (with an uppercase "D" as a member of the Deaf Community), deaf, or hard of hearing is indicative of cultural identity (Lane, 1992; Padden & Humphries, 1988, 2005).[1] Membership in the Deaf Community is determined more by shared language and worldviews rather than by results of audiometric tests. For example, people may lose their hearing with age. These people, while deaf by the audiometric definition, would not share the culture of the Deaf Community.

In the United States, the language of the Deaf Community is American Sign Language (ASL). Other countries and locales have their own native sign languages shared by members of Deaf Communities in those areas. Interestingly, these signed languages are not based on the spoken languages of the region. People are often surprised to learn that ASL is more similar to French Sign Language, from which it originated (Lane, 1984), than it is to British Sign Language. Beginning with the seminal work of Ursula Bellugi and colleagues in the 1970s, linguists, cognitive psychologists, and brain researchers have studied native sign languages and their users for clues as to the origin of language and the biological nature of language (Emmorey & Lane, 2000; Klima & Bellugi, 1979; Erard, 2005).

Cochlear implants are a medical intervention that has received much attention in the last couple of decades. The decision as to whether or not to have a cochlear implant is often a complex one, as was explored in the movie *Sound and Fury* (Aronson, 2000; see also Hyde & Power, 2006). From the standpoint of a user, an implant is not the same as perfect hearing, but does allow the user to hear sounds and, with training, may greatly aid in the perception of speech (Chorost, 2005).

HEARING AND LANGUAGE

Some, but not all, deaf and hard of hearing individuals use sign language. The type of signed language depends on the user's life experiences. Deaf children born to deaf parents, regardless of severity of hearing loss, will generally acquire a sign language, such as ASL, natively as hearing children of hearing parents acquire spoken language. People who lose their hearing late in life generally will not master a sign language. In between, there are many variations. Deaf signers may be exposed to a native sign language as adults and as a result, may only acquire partial mastery (Newport, 1990). Other deaf signers will be exposed primarily to manual forms of English (or other spoken languages), rather than natural sign languages. Many schools that use sign

[1]Terms such as hearing-impaired and deaf-mute are generally considered to have negative connotations. For a discussion of this, see "What is Wrong with the Use of these Terms: 'Deaf-mute,' 'Deaf and dumb,' or 'Hearing-impaired'?" by the National Association of the Deaf. (Available at http://www.nad.org/site/pp.asp?c=foINKQMBF&b=103786). The style manual of the American Psychological Association (APA) recommends use of nondiscriminatory language in all publications (see http://www.apastyle.org/disabilities.html).

language in the classroom do not use a natural sign language, but rather a representation of the spoken language that is signed. Forms of signed English borrow signs from ASL, but these signs are produced in English word order rather than using ASL sentence structures (for an extended discussion, see Lane, Hoffmeister, & Bahan, 1996). Children attending schools that use some form of signed English may not be exposed to ASL. Moreover, schools that educate students using an oral approach, in which speech is the primary means of communication in the classroom, also may not be exposed to ASL.

A person who is profoundly deaf from birth may have difficulty acquiring mastery of the spoken language, be it presented auditorily or in print. It is not surprising that someone who has never heard speech will have difficulty perceiving or producing it. In a large-scale study of deaf and hard of hearing children attending schools using an oral approach to education, Conrad (1979) reported that profoundly deaf children rarely acquired sufficient lipreading skills to allow easy participation in conversations. He found, on average, that these children (with hearing loss greater than 85 dB) could only comprehend about 25% to 28% of the words through lipreading that they could comprehend through reading. Even among these orally educated students, fewer than 20% had speech that was rated even fairly easy to understand. While the statistics are better for hard of hearing children, Conrad found that even these students (with hearing loss less than 65 dB) could only comprehend about 36% of the words through lipreading that they could comprehend through reading. Nearly 85% of these hard of hearing students, however, had speech that was rated at least fairly easy to understand.

Perhaps more surprising may be the fact that many deaf and hard of hearing individuals have difficulty with reading (Conrad, 1979; Gallaudet Research Institute, 2003). To understand this, it is necessary to realize that reading is based on the underlying spoken language. Spoken languages are composed of sounds, linguistically defined as "phonemes." These phonemes correspond, albeit not always in a one-to-one relationship in English, to letters or letter combinations. This is true of all alphabetic languages. Learning to read is generally considered to be learning to map the print onto the spoken language the person already knows (Brady & Shankweiler, 1991). In addition, speech plays a critical role in the short-term memory processes that serve understanding of grammar and text comprehension (Lichtenstein, 1998). In the case of deaf readers, however, it cannot be taken for granted that reading will build on a firm understanding of the structure of the spoken language, but there are no absolutes. Some prelingually, profoundly deaf children become excellent readers, while some with lesser degrees of hearing loss do not (Conrad, 1979; Hanson, 1989).

Interestingly, research has recently turned to an examination of signed languages as an influencer in the development of skilled reading. It has long been known that deaf children of deaf parents, on average, acquire greater mastery of reading and writing than deaf children of hearing parents. Is this due to early exposure to sign language? While a number of both intellectual and societal factors have been considered as contributors to this disparity, interest has focused on the issue of language. Evidence is now emerging as to the important role that early mastery of a sign language can have on second language learning for deaf students as they acquire reading and writing skill (Padden & Hanson, 2000; Padden & Ramsey, 2000).

In short, for any deaf or hard of hearing individual, language experience cannot be assumed. That individual may or may not sign, speak clearly or lipread well, or have reading skills consistent with those of the hearing population. This knowledge has implications for designers who seek to address the needs of deaf and hard of hearing users.

DISPLAY TECHNOLOGIES

In many ways, computers and other technologies have proven to be of great benefit to deaf and hard of hearing users. The largely visual nature of information on the Internet makes this information accessible to deaf and hard of hearing users. Instant messaging and e-mail facilitate communication with deaf and hearing family, friends, and coworkers, while network-connected wireless devices such as PDAs and pagers are seen as lifelines that can be used in place of cell phones. As would be suspected from the previous discussion, however, effective interfaces for these and other technologies for use by deaf and hard of hearing individuals must take into account both sensory and language considerations. In particular, the increasing reliance on sound and speech interfaces to convey information can have serious consequences for individuals who have a hearing loss.

Audible Signals

Sounds have become increasingly popular in computer interfaces. They have long been used to convey information about new messages and have become popular as problem alerts, such as when an error has been committed. These sound events are considered attention-grabbing events for users whose visual attention may otherwise be engaged. For any user who has a hearing loss, however, sounds will be a problem.

A number of considerations can help provide the necessary visual support for a user who is deaf or hard of hearing (Vanderheiden, 1994). It is necessary to provide visual forms for all auditory information. Critically, these visual cues should be sufficiently noticeable so that they catch the attention of a person who may not be looking directly at the computer screen. Operating systems have features that can provide such visual alerts. For example, Windows® has accessibility features that allow users to set up their system to have captions or visual warnings displayed for sound events. These are helpful, although they may not give the full range of information carried by a sound event. For example, the meaning of a sound event may differ based on the tone of the signal or when the sound is produced. While visual captions and warnings may alert a deaf user that a sound event has occurred, they will be unable to convey these more subtle distinctions. It is important for designers to give careful consideration to sound events to ensure that crucial information is available by a nonauditory means for deaf and hard of hearing users.

Multimodal interfaces, as the name implies, are designed to support a range of perceptual capabilities. In theory, this would

seem ideal for users who are deaf or hard of hearing as visual alternatives to auditory materials should be available. Multimodal interfaces, however, do to always present all information on both modalities. The emphasis in many multimodal interfaces is representing information via speech that would otherwise be conveyed by print or some other visual means. To the detriment of deaf and hard of hearing users, often, less attention is given to ensuring that all auditory material be visually conveyed as well.

Multimedia Interfaces

Multimedia uses a combination of text, sound, pictures, animation, and video to present information (see Sutcliff, chapter 20, "Multimedia User Interface Design," this volume). Traditionally, games and educational software have exploited the richness of multimedia, but the advent of high-speed Internet communications has enabled the use of multimedia for a number of engaging applications on the web. For the present discussion, consideration will be given to multimedia presentations as they may impact access for deaf and hard of hearing users.

Multimedia material is inherently sensory. The technology offers eye-catching visual displays and attention-grabbing sound effects. To the extent that the information conveyed visually and auditorally is the same, information can be reinforced for users who have both channels available to them. To the extent that different information is presented in the two channels, however, users who have a functional loss of one of the channels will not have full access to that information. As it relates to the present discussion, this means that any information that is carried solely by sound or speech will be unavailable to deaf and hard of hearing users. Accessibility guidelines for multimedia products and web pages require equivalent visual presentations (e.g., see Brewer & Dardailler, 1999; U.S. General Services Administration, n.d.).

Consider the growing popularity of video on the web. Video material is an extremely effective way to convey information and younger generations of computer users have come to expect video to be part of their computer experience. For deaf and hard of hearing users, the voice-over that is common in video will be inaccessible. Words spoken by persons in view of the camera also are largely inaccessible. Even people skilled at lipreading cannot lipread video conversations that have poor lighting or poor resolution, or speakers who turn away from the camera. Additionally, the video may present music or sound events (e.g., doorbells or animals noises) that contribute significantly to events on the video. These, too, are unavailable to deaf and hard of hearing users. Technologies such as captioning and sign-language translation exist to provide alternative presentations of sound events and speech. Designers wishing to make their applications universally accessible should consider these alternatives and incorporate them into their applications.

Captioning. Captioning provides a print alternative to speech and sound events. It is much like subtitling, except that it is specifically designed for deaf and hard of hearing users and, thus, will include comments in the captioning about sounds (e.g., "<music playing>" or "<sounds of child crying>") that may not be included in subtitles of foreign language videos. Captioning of certain television programming is mandated in the United States by the Federal Communications Commission and is considered to be beneficial to large number of users, not only those with hearing loss. It has been shown, for example, to improve reading abilities of children and to benefit second language learners. Designers who use multimedia materials have an obligation to their full audience to provide captioning of audio and video materials. The listing of a number of resources for captioning software is available at the Closed Captioning \website (n.d.).

As might be anticipated from the previous review of reading levels of deaf and hard of hearing users, there has been some controversy about what language level should be used for captioning. Simply put, the issue revolves around the question of whether captions should be verbatim transcripts or simplified captioning should be provided. Verbatim captioning is generally preferred by users themselves (NIH, 2002) and is often cited as having the potential to improve reading skills (Steinfeld, 2001).

Signing. For sign language users, there are sign language alternatives to captioning. These sign interfaces have been defined as ways of representing signed languages on a computer such that signing can be stored, displayed, and manipulated to facilitate computer interaction (Frishberg, Corazza, Day, Wilcox, & Schulmeister, 1993). It should be noted that these interfaces have often been employed not only for making audio and speech materials accessible to deaf and hard of hearing users, but they have also been used for language learning by both deaf and hearing people. A number of software applications have been developed that use sign language as a means of teaching reading skills and writing to deaf signers or teaching sign language skills to individuals wishing to learn to sign.

Importantly, software applications that purport to use "sign language" or ASL differ in significant ways in the language that is being used. Notably, many of these applications use fingerspelling. Fingerspelling is not a natural signed language, but rather is a derivative of print. Specifically, in fingerspelling, there is one handshape for each letter of the alphabet and words are spelled out letter by letter on the hands. Thus, the word "language" would be spelled out by eight distinct handshapes spelling L-A-N-G-U-A-G-E. Other sign language interfaces use a form of signed English, rather than the native sign language. For young children and others not fluent in English, interfaces that use fingerspelling or signed English transliterations of text may not meet their needs.

In many cases, the goal when using sign interfaces is to provide ASL translations. Given the present state of the art, automatic translation from English to ASL is not possible; however, many current efforts are directed at facilitating this translation, acknowledging the need for translation into ASL. In what follows, a few of the options for sign language presentation of audio and multimedia will be discussed.

Ideal in many respects would be to have a live person signing an ASL version of print and multimedia materials. Hanson and Padden (1989, 1990) used videodisc technology in the earliest computer-based attempt at a bilingual ASL/English approach to reading and writing instruction for signing children. The work combined ASL video and the translated English text on one

screen. In this and other language learning situations where the users are young children still developing both ASL and English skills, this use of live signing has been particularly effective (Frishberg et al, 1993).

The advent of high-speed networks and Internet video has created opportunities for web applications that use live signing interfaces. For example, classroom applications have been developed (e.g., see King, 2000; Laurent Clerc National Deaf Education Center, n.d.; for a demo, see "Sample Web Page," Gallaudet University, n.d.). Additionally, video blogs have created the opportunity for blogs to be signed by the blogger, rather than written (Lamberton, 2005), and video e-mail allows signers to communicate through signing by creating video recordings to be transmitted as e-mail messages (e.g., see "Road Runner Video Mail," n.d.; also, "Vibe Video Mail," n.d.).

Although ideal from the language perspective, other constraints may argue against the use of live signing in an interface. Lack of access to high-speed networks for video transmission may be an issue, but often the problem is the desire for automatic translation of software and web content that live signing does not provide. Live signing requires the prior recording of the signed material. Once recorded, changes to the video require a new recording. Because of this, live signing is not practical in applications that require that sign versions of audio or multimedia be created in real time. Short of having an interpreter doing the translations, live signing is not possible in these situations.

One approach to automatic sign presentation is what might be called "concatenated signing" (e.g., see "iCommunicator," n.d.; "Signtel," n.d.). With this, a software program is used to create word strings or sentences by concatenating signs produced by a live signer. The program starts with a vocabulary of stored individual signs. These signs can be strung together to form phrases and sentences. To prevent a jerky appearance that would occur through the simple production of a list of signs, algorithms are used to smooth the transitions between these signs. While these transitions are not as natural as live signing, the signing is legible and suggests an interesting approach to sign language interfaces.

A technique that has generated much interest in recent years is signing avatars (e.g., see Cox et al., 2002; Karpouzis, Caridakis, Fotinea, & Efthimiou, in press; Kennaway, 2002; Sims, 2004). These avatars use virtual-reality techniques to produce animated signing. The specific techniques differ for the various avatars, but central to all is that they display computer generated signing. Some systems are able to display not just hands, but full signers, so that facial expressions as well as hand movements are shown. See, for example, the SigningAvatar® shown in Fig. 45.1 that illustrates the sophistication of these animations.

Signing avatars have been used in education and have potential for applications such as translation of web pages, television programs, and conversational dialog. Chief among the virtues is that, unlike natural (live) sign language applications that are limited to prerecorded materials, avatars can generate signed versions of English words "on the fly." They also have the advantage of not requiring large downloads for web usage.

While the avatars can sign ASL when preprogrammed, the language translation work needed for automatic translation into ASL (or other signed language) is not ready to support this rendering on the fly. A current research focus in avatar work is on natural language translation and exploring techniques to display native sign languages by avatars (e.g., see Huenerfauth, 2005).



FIGURE 45.1. An ASL sign avatar. The full animation of this avatar can be viewed on http://www.vcom3d.com/ASL.htm. © Vcom3D, Inc., 2004. All rights reserved.

INPUT TECHNOLOGIES

Standard keyboard and mouse input technologies offer no barriers to users who have a hearing loss. The emergence of new interfaces, however, presents alternatives that will impact the way in which we are able to interact with computers. We consider here both speech and gestural interfaces as they may influence interactions for deaf and hard of hearing users.

Speech Interfaces

Conversational speech interfaces have appeal as a natural means of interacting with computers (see Karat, Vergo, Nahamoo, Lai, & Yankelovich, chapter 19, "Conversational Speech Interfaces and Technologies," this volume). These interactions can range from simple, even one-word commands to full dictation of documents and user collaborations. For deaf and hard of hearing individuals, such interactions require consideration of user needs. First, these interactions can involve speech output which, as already discussed, requires a visual display alternative. Second are problems with speech input. Some deaf and hard of hearing individuals utilize speech and would be interested in taking advantage of speech input. Speech recognition, however, may be more problematic for these speakers than for hearing speakers.

As mentioned briefly in the previous paragraph, the speech of deaf and hard of hearing individuals is not always highly intelligible to hearing listeners. Since speech recognition engines can be trained to the voice and pronunciations of an individual speaker, however, couldn't a recognizer be trained to understand the speech of an individual deaf speaker, even if the

speech is not completely intelligible to listeners? The difficulty is that recognizers require consistent speech. Research has shown that the speech of deaf and hard of hearing speakers often is more variable than the speech of hearing speakers (McGarr, 1987; McGarr & Lofqvist, 1988). For example, there is more acoustic variation in the pronunciation of a single phoneme by a deaf or hard of hearing speaker than there is in the pronunciation of that phoneme by a hearing speaker. Hearing listeners are very tolerant of this variability; speech recognizers are less so. Thus, deaf and hard of hearing users who many wish to use their speech for input may well find recognition less accurate than it is for hearing speakers.

Speech interfaces are often seen as useful alternatives to visual interfaces in situations when computers or keyboards are not available (see Karat et al., chapter 19, "Conversational Speech Interfaces and Technologies," this volume). For hearing users, this alternative of a speech interface may be highly desirable. Many deaf and hard of hearing users, however, will not be interested in speech interfaces; others may experience difficulties in using them. Although situational demands may sometimes dictate the use of conversational interfaces, care must be taken that outside of the situational context, there exists a means for deaf and hard of hearing users to access the same information that hearing users access.

Sign Interfaces

Designed specifically for deaf and hard of hearing users, sign recognition is a specific and complex subset of gesture recognition technologies having the goal of automatically converting signed language to text or speech. These technologies have been investigated for a number of years, primarily to address the need to facilitate communication between signers and nonsigners. They also have the potential to provide an alternative means of natural language input to computers.

The task of recognizing full ASL or other natural sign languages is a difficult problem. The first reason is that an individual sign varies depending upon its context. Take, for example, the sign for the word GIVE. This sign has a specific shape combined with a variable movement. The movement reflects who is giving and to whom. I GIVE-TO YOU, YOU GIVE-TO ME, I GIVE-TO HIM, HE GIVES-TO ME, and I GIVE-TO ALL-OF-YOU each have a different movement that indicates subject and object. A second reason for difficulty in sign-language recognition is that several pieces of linguistic information are produced in parallel. For example, facial expression carries critical grammatical information. Thus, a full language recognizer needs to recognize not only the hand gestures of ASL, but must also recognize certain facial information relevant to the grammar. Such facial elements include eyebrow position, eye gaze, and mouth movements.

As with interfaces that produce signs, many systems that purport to perform sign-language recognition deal with recognition of fingerspelling handshapes rather than recognition of ASL sentences or even ASL signs. Using fingerspelling handshapes rather than ASL signs certainly constrains the size of the problem. Because there are only 26 handshapes in the English alphabet, this represents a much more manageable problem space than the recognition of full ASL signing. While it presents an alternative to keyboard typing of words, it doesn't provide the type of natural language interaction with computers that is afforded by speech interfaces.

Techonologies for recognizing signs have tended to use either instruments worn by the signer or computer vision techniques. The first if these approaches has the signer wear a specially designed glove or sensors placed on their joints that allow a computer to track movement (e.g., see Braffort, 1996; Fang, Gao, & Zhao, 2003; Hernandez-Rebollar, Lindeman, & Kyriakopoulos, 2002; Kadous, 1996; Wang, Gao, & Ma, 2002). In contrast, computer vision techniques use cameras to provide input to a computer about a signer's movements and facial gestures (e.g., see Brashear, Starner, Lukowicz, & Junker, 2003; Kadous, 1996; Lee, et al., 2005; Vogler & Metaxas, 2001). These inputs are then analyzed using a variety of techniques such as neural networks or Hidden Markov Models that then recognize the sign.

Critically, however, this recognition does not do language translation. Thus, the output will be a one-to-one mapping of a sign into print or speech. As mentioned with the avatar work presented earlier, the ASL/English rules that would be needed for such translation are not developed to a state where automatic translation can occur; however, advances are being made. Recently, for example, Hernandez-Rebollar (2005) presented work designed to translate signed input into English phrases. In that work, the translation was enabled by having a limited number of phrases that the system could recognize, thus constraining the problem space.

The current state of the art for sign recognition is not as advanced as speech recognition. As researchers continue to work on the problem, however, advances can be expected. For signers, it might be the case that conversational sign interactions will one day be possible, much as speech recognition now allows speakers to benefit from conversational speech interactions.

TECHNOLOGY AND COMMUNICATION

Deaf and hard of hearing individuals have difficulties not only with computer interfaces, but also experience significant difficulties in certain communication situations. Telephone conversations, as well as one-on-one conversations, group discussions, and presentations or classroom lectures, are all problematic. No discussion of HCI for deaf and hard of hearing individuals would be complete without at least a brief mention of technologies that have been developed to aid in these situations.

Telephones have long been a source of difficulty for deaf and hard of hearing people. Alexander Graham Bell was a teacher of deaf students and was married to a woman who was deaf. His interest in finding improved ways to communicate with deaf speakers lead to his invention of the telephone. Ironically, however, over the years, telephones created a number of barriers for deaf and hard of hearing individuals in the workplace and other situations. To overcome these barriers, a number of assistive devices have been developed (Lazzaro, 1993). Amplification and adapters exist for many phones that will allow hard of hearing individuals or people with cochlear implants to hear phone conversations. Teletype devices (TTYs and TDDs) as well as some

computer applications allow deaf users to type conversations that are carried over phone lines.

Operator-assisted relay services have been established to enable conversations between deaf and hearing individuals. The relay personnel serve as a bridge between the two conversation participants, translating typed information into speech for the hearing participant and translating speech into written text for the deaf participant. Hard of hearing speakers can speak directly with others using a captioning service that provides, nearly in real time, a printed transcript of the conversation to support hard of hearing users (CapTel, 2005). Various means of enabling signed conversations over the telephone have been explored over the years, but the advent of video phones has now made feasible the option of signed phone conversations.

The current prevalence of conference calls in the workplace has created a new set of problems for deaf and hard of hearing workers. Similarly, classroom situations or lecture presentations also create significant difficulties for deaf and hard of hearing attendees. Even individuals skilled in lipreading have difficulty in these situations because the speaker's face is rarely visible with sufficient resolution for lipreading. Sign language interpreting and captioning are two means of providing accommodation for participants unable to hear or understand the speech in these situations.

Sign language interpreters allow signers to participate in meetings by providing real-time translation of the spoken conversation into sign. This is a two-way translation service, such that the signing participate is also able to participate by having the interpreter speak their signed utterances. Remote interpreting is a technology that addresses the problem of a shortage of skilled interpreters, particularly in some locations. The interesting aspect of this is that the interpreter need not be present at the location of any of the conference participants. The deaf or hard of hearing participant views the remotely located interpreter on a computer display or TV screen. The interpreter listens to hearing participants by telephone and provides sign-language interpreting for the deaf participant. The interpreter also voices what the deaf person signs for hearing participants. This remote interpreting can be used similar to the telephone relay service, with participants in different locations (Video Relay Service, or VRS), or in situations where the participants are located in the same room (Video Relay Interpreting, or VRI).

Communication Access Real-time Translation (CART) provides real time captioning, enabling discussions and presentations to be transcribed into text for deaf and hard of hearing participants. Typically, this is done by a person who creates the captions. As with sign language interpreting, the captioner can be physically present or remotely located, listening to the conversation by phone and transmitting the text via a network to a computer screen or projected display. The deaf participant types their questions or comments. That input is then read aloud for the other participants, either by a captioner or using computer text-to-speech technology (e.g., see Caption First, n.d.; Viable Technologies, n.d.)

The ability to automatically transcribe speech was envisioned more than 100 years ago as a technology that held great promise for deaf and hard of hearing people, even though such technology was hardly imaginable at the time (Fay, 1883*)*. As speech-recognition systems have matured over the last quarter century, a number of applications designed for deaf users have been explored (Bain, Basson, Faisman, & Kanevsky, 2005; Stinson & Stuckless, 1998).

The Liberated Learning Project is one example of using automatic speech recognition in the classroom (Bain, Basson, & Wald, 2002). In this effort, the classroom teacher speaks into a microphone that transmits his or her speech to a computer to perform the recognition. The transcript of the lecture is displayed in close to real time on a screen at the front of the classroom. Shown in Fig. 45.2 is a classroom situation for the Liberated Learning Project.



FIGURE 45.2. An example classroom lecture transcribed in real time. From Bain et al. (2005); republished with permission.

While recognition technologies have improved over the years, they still do not attain 100% correct performance for dictation. To provide students with accurate transcripts for lecture notes, the professors who participate in the Liberated Learning Project review the transcript for inaccuracies and make corrections after a lecture is completed. The corrected transcripts are then made available to students (Bain et al., 2002). Other efforts that use automatic speech recognition for captioning use different methods for correcting errors in transcripts. These other methods include having a trained speaker "shadow" the speech to produce more reliable recognition, and/or having a person correcting errors in real time (e.g., see Bain et al., 2005; Robson, 2001; Viable Technologies, n.d.).

CONCLUSIONS

Computer technologies are very important in the lives of many deaf and hard of hearing people. These technologies can ease communication between coworkers, friends, family members, neighbors, and a variety of services. These technologies have also provided access for deaf students, employees, and individuals to have full access to rich multimedia services, shopping, news, and general information. In short, barriers long encountered by deaf and hard of hearing people are now being overcome through the use technology.

Deaf and heard of hearing users represent a population that, in itself, is diverse in both hearing and language experiences and skills. Applications that offer flexibility of language (e.g., captioning, signed English, or ASL) will be most accessible to users who have a hearing loss. At a minimum, however, developers and designers need to ensure that information is not carried by the auditory channel alone. Paramount is the need to ensure that any audible information, be it a sound alert, speech prompt, or other auditory event, have a visual counterpart. Such considerations are not only good design, but are also mandated by a growing number of regulations worldwide. The ability of everyone to participate in our increasingly technological society is crucial.

ACKNOWLEDGMENTS

I'd like to thank Al Noll, Debra Noll, and John Richards for valuable comments on early drafts of this chapter.

References

Aronson, J. (2000). *Sound and Fury* [Motion picture]. United States: New Video Group.

Bain, K., Basson, S., & Wald, M. (2002). Speech recognition in university classrooms: Liberated learning project. In *Proceedings of the 5th International ACM SIGCAPH Conference on Assistive Technologies* (*ASSETS'02*), (pp, 192–196). New York: ACM.

Bain, K., Basson, S., Faisman, A., & Kanevsky, D. (2005). Accessibility, transcription, and access everywhere. *IBM Systems Journal, 44*(3), 589–605.

Brady, S., & Shankweiler, D. (1991). *Phonological processes in literacy: A Tribute to Isabelle Y. Liberman*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Braffort, A. (1996). A gesture recognition architecture for sign language. In *Proceedings of the 2nd International ACM SIGCAPH Conference on Assistive Technologies* (ASSETS'96), 102–109.

Brashear, H., Starner, T., Lukowicz, P., & Junker, H. (2003). Using multiple sensors for mobile sign language recognition. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, 45–52.

Brewer, J. & Dardailler, D. (1999). *Web Content Accessibility Guidelines 1.0*. Retrieved December 7, 2005, from University of Wisconsin-Madison, Trace Research and Development Center website: http://www.w3.org/TR/WAI-WEBCONTENT/

CapTel. (n.d.). *Introducing the Captioned Telephone.* Retrieved December 7, 2005, from http://www.captionedtelephone.com/about captel.phtml

Caption First. (n.d.). *Overview of our services.* Retrieved January 20, 2006, from http://www.captionfirst.com/overview.htm

Chorost, M. (2005). *Rebuilt: How becoming part computer made me more human.* New York, NY: Houghton Mifflin Company.

Closed Captioning Web. (n.d.) *Closed Captioning Web: Software Links.* Retrieved December 7, 2005, from http://www.captions.org/softlinks.cfm

Conrad, R. (1979). *The deaf schoolchild.* London: Harper & Row.

Cox, S., Lincoln, M., Tryggvason, J., Nakisa, M., Wells, M., Tutt, M., et al. (2002). TESSA, a system to aid communication with deaf people. In *Proceedings of the 5th International ACM SIGCAPH Conference on Assistive Technologies* (*ASSETS'02*). (pp. 205–212). New York: ACM.

Emmorey, K., & Lane, H. (2000). *The signs of language revisited: An anthology to honor Ursula Bellugi and Edward Klima*. Mahwah, NJ: Lawrence Erlbaum Associates.

Erard, M. (2005). The birth of a language. *New Scientist, 188*(2522), 46–49.

Fang, G., Gao, W., & Zhao, D. (2003). Large vocabulary sign language recognition based on hierarchical decision trees. In *Proceedings of the 5th international conference on Multimodal interfaces*. (pp. 125–131). New York: ACM.

Fay, E. A. (1883). The glossograph. *American Annals of the Deaf, 28*, 67–69.

Frishberg, N., Corazza, S., Day, L., Wilcox, S., & Schulmeister, R. (1993). Sign language interfaces. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. (pp. 194–197). New York: ACM.

Gallaudet Research Institute. (2003). *Literacy and deaf students*. Retrieved December 7, 2005, from http://gri.gallaudet.edu/Literacy/

Gallaudet University. (n.d.). *Sample web page with embedded real media video*. Retrieved January 20, 2006, from http://academic.gallaudet.edu/pages/iced2000/real/stellaluna_smil.html

Hanson, V. L. (1989). Phonology and reading: Evidence from profoundly deaf readers. In D. Shankweiler & I. Y. Liberman (Eds.), *Phonology and reading disability: Solving the reading puzzle* (pp. 69–89). Ann Arbor: University of Michigan Press.

Hanson, V. L., & Padden, C. A. (1989). The use of interactive video for bilingual ASL/English instruction of deaf children. *American Annals of the Deaf, 134*, 209–213.

Hanson, V. L., & Padden C. A. (1990). Bilingual ASL/English instruction of deaf children. In D. Nix & R. Spiro (Eds.), *Cognition, education, and*

multimedia: Exploring ideas in high-technology (pp. 49–63). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hernandez-Rebollar, J. L. (2005). Gesture-drive American Sign Language phraselator. In *Proceedings of the 7th International ACM Conference on Multimodal Interfaces* (*ICMI'05*). (pp. 288–292). New York: ACM Press.

Hernandez-Rebollar, J.-L., Lindeman, R. W., & Kyriakooulos, N. (2002). A multi class pattern recognition system for practical fingerspelling translation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces* (ICMI'05) (pp. 185–190). Washington, DC: IEEE Computer Society.

Huenerfauth, M. (2005). Representing coordination and non-coordination in an American Sign Language animation. In *Proceedings of the 7th International ACM SIGACCESS Conference on Assistive Technologies* (*ASSETS'05*). (pp. 44–51). New York: ACM Press.

Hyde, M., & Power, D. (2006). Some ethical dimensions of cochlear implantation for deaf children and their families. *Journal of Deaf Studies and Deaf Education, 11*(1), 102–111.

iCommunicator. (n.d.). *iCommunicator.* Retrieved January 4, 2006, from http://www.myicommunicator.com/

Kadous, M. W. (1996). Machine recognition of Auslan signs using Powergloves: Towards large-lexicon recognition of sign languages. In L. Messing (Ed.), *Proceedings of WIGLS, The Workshop on the Integration of Gestures in Language and Speech, Wilmington Delaware*, 165–174.

Karpouzis, K., Caridakis, G., Fotinea, S.-E., & Efthimiou, E. (in press). Educational resources and implementation of a Greek sign language synthesis architecture. *Computers in Education.*

Kennaway, R. (2002). *Synthetic animation of deaf signing gestures*. Gesture and Sign Language in Human-Computer Interaction: International Gesture Workshop (GW 2001, LNAI 2298), 146–157.

King, C. (2000). *Online learning at Gallaudet University.* Retrieved January 20, 2006 from Gallaudet University website: http://academic.gallaudet.edu/pages/iced2000/iced2000_GUonlinelearning.PDF

Klima, E. S., & Bellugi, U. (1979). *The signs of language*. Cambridge, MA: Harvard University Press.

Lamberton, J. (2005). *Jason Lamberton's video blog.* Retrieved January 20, 2006, from http://video.google.com/videoplay?docid=6012463606293405795&q=gallaudet+university

Lane, H. (1984). *When the mind hears: A history of the deaf*. New York: Random House.

Lane, H. (1992). *The mask of benevolence: Disabling the deaf community.* New York: Alfred A. Knopf.

Lane, H., Hoffmeister, R., & Bahan, B. (1996). *A journey into the deaf-world*. San Diego, CA: Dawn Sign Press.

Laurent Clerc National Deaf Education Center, Gallaudet University. (n.d.). *Shared reading project: Chapter by chapter—The Thinking Reader.* Retrieved January 20, 2006, from http://clerccenter.gallaudet.edu/Literacy/programs/chapter.html

Lazzaro, J. J. (1993). *Adaptive technologies for learning and work environments.* Chicago, IL: American Library Association.

Lee, S., Henderson, V., Hamilton, H., Starner, T., Brashear, H., & Hamilton, S. (2005). A gesture-based American Sign Language game for deaf children. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (*CHI '05*). (pp. 1589–1592). New York: ACM Press.

Lichtenstein, E. H. (1998). The relationships between reading processes and English skills of deaf college students. *Journal of Deaf Studies and Deaf Education, 3*(2), 80–134.

McGarr, N. S. (1987). Communication skills of hearing-impaired children in schools for the deaf. *ASHA Monographs, Oct*(26), 91–107.

McGarr, N. S., & Lofqvist, A. (1988). Laryngeal kinematics in voiceless obstruents produced by hearing-impaired speakers., *Journal of Speech and Hearing Research, 31*(2), 234–239.

Mitchell, R. E. (2005, February 15). *Can you tell me how many deaf people there are in the United States?* Retrieved December 7, 2005, from Galludet Research Institute, Graduate School and Professional Programs website: http://gri.gallaudet.edu/Demographics/deaf-US.php

Mitchell, R. E. (2006). How many deaf people are there in the United States? Estimates from the survey of income and program participation. *Journal of Deaf Studies and Deaf Education, 11*(1), 112–119.

National Institutes of Health, National Institute on Deafness and Other Communication Disorders. (2002). *Captions for deaf and hard of hearing viewers*. NIH Publication No. 00-4834. Retrieved December 7, 2005, from http://www.nidcd.nih.gov/health/hearing/caption.asp#edit

National Institutes of Health, National Institute on Deafness and Other Communication Disorders. (2004). *Statistics about hearing disorders, ear infections, and deafness*. Retrieved December 7, 2005 from http://www.nidcd.nih.gov/health/statistics/hearing.asp

Newport, E. (1990). Maturational constraints on language learning. *Cognitive Science, 14*, 11–28.

Padden, C. A., & Hanson, V. L. (2000). Search for the missing link: The development of skilled reading in deaf children. In K. Emmorey & H. Lane (Eds.), *The signs of language revisited: An anthology to honor Ursula Bellugi and Edward Klima* (pp. 435–447). Mahwah, NJ: Lawrence Erlbaum Associates.

Padden, C., & Humphries, T. (1988). *Deaf in America: Voices from a culture*. Cambridge, MA: Harvard University Press.

Padden, C., & Humphries, T. (2005). *Inside deaf culture*. Cambridge, MA: Harvard University Press.

Padden, C., & Ramsey, C. (2000). American Sign Language and reading ability in deaf children. In C. Chamberlain, J. Morford, & R. Mayberry (Eds.), *Language acquisition by eye* (pp. 165–189). Mahwah, NJ: Lawrence Erlbaum Associates.

Road runner video mail. (n.d.). Retrieved January 20, 2006, from http://vmail.vibephone.com/vm/vm_player?vmfile=4dbc51d033b4cdbac583bf8ec3ef9cd3&vmpid=1007

Robson, G. (2001). *Can realtime captioning be done using realtime voice recognition systems?* Retrieved December 7, 2005, from http://www.robson.org/capfaq/online.html#VoiceRecognition

Signtel, Inc. (n.d.). *Signtel.* Retrieved December 7, 2005, from http://www.signtelinc.com/main.htm

Sims, E. (2004). *Using emerging visualization technologies to provide sign language access to the Web*. Retrieved January 20, 2006 from http://www.w3.org/WAI/RD/2003/12/Visualization/VCom3D.html

Steinfeld, A. (2001). *The case for real time captioning in classrooms*. Retrieved December 7, 2005, from http://www.cartinfo.org/steinfeld.html

Stinson, M., & Stuckless, R. (1998). Recent developments in speech-to-print transcription systems for deaf students. In A. Weisel (Ed.), *Issues unresolved: New perspectives on language and deaf education* (pp. 126–132). Washington, DC: Gallaudet University Press.

U.S. General Services Administration, Office of Governmentwide Policy, IT Accessibility & Workforce Division. (n.d.) *Section 508.* Retrieved January 20, 2006 from http://section508.gov/

Vanderheiden, G. C. (1994). *Application software design guidelines: Increasing the accessibility of application software for people with disabilities and older users*. Retrieved January 20, 2006, from University of Wisconsin-Madison, Trace Research and Development Center website: http://trace.wisc.edu/docs/software_guidelines/software.htm

Viable Technologies. (n.d.). *Frequently asked Questions*. Retrieved December 7, 2005, from http://www.viabletechnologies.com/faq.php

Vibe video mail. (n.d.). Retrieved January 20, 2006, from http://www.vibephone.com/vsg/htdocs/products/video-mail/index.jsp

Vogler, C., & Metaxas, D. (2001). Framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding* (81), 358–384.

Wang, C., Gao, W., & Ma, J. (2002). A real-time large vocabulary recognition system for Chinese Sign Language. In I. Wachsmuth and T. Sowa (Eds.), *International Gesture Workshop, Springer Lecture Notes in Artificial Intelligence*, (Vol. 2298, pp. 86–95). New York, NY: Springer-Verlag.

Part

VI

SECTION A

THE DEVELOPMENT PROCESS REQUIREMENTS SPECIFICATION

·46·

USER EXPERIENCE AND HCI

Mike Kuniavsky
ThingM Corporation

Introduction **898**
The Boundaries of User Experience **898**
UX Is Context 898
Garrett's Elements 898
The Organizational Experience **899**
The 1927 Ford Model T 900
A Children's Art Product Manufacturer Website 900
The User View **901**
The User Experience of Products 901
Affect 901
Value 901
The User Experience of Organizations 902
Brand 902
Relationships 904
Examining the User Experience **905**
Identifying Organizational Goals 905
Identify stakeholders 905
Collect stakeholder goals 906
Prioritize organizational goals 906
A rapid technique: project history 906
Field Observation 907
Find key informants, schedule research 908
Narrow the focus 908
User interactive observation 909
Use multiple researchers and analyze collaboratively 909
Validation 909
Focus Groups 909
Prepare 910
Make a schedule 910
Pick an audience 910
Develop discussion topics 911
Write a discussion guide 911
Analyze results 911
Manage With User Experience **911**
Agile User Experience Development 912
Iterative development 912
Risk-driven and client-driven 912
Timeboxing 913
Adaptive development and evolutionary requirements 913
Introducing User Experience Into an Existing Process 913
Get a senior manager champion 913
Work within existing processes 913
Make small, but well-publicized changes 914
Make developers' lives easier with user experience 914
Conclusion **914**
References **915**

INTRODUCTION

The goal for this chapter is to introduce concepts and techniques that help structure the application of HCI in a real-world environment by examining the larger context in which HCI happens and by using that context as the basis for the design of user experiences.

Understanding the broader factors that influence the user experience is as important for creating successful HCI systems as thoroughly understanding the cognitive science behind the user's actions. Company goals, economic relationships, emotional responses, and social interactions can overwhelm behavioral and perceptual responses of consumers. Although intensive research is currently investigating some of these ideas, the majority of firsthand experience of and thinking about designing experiences under such pressures has happened in the consumer marketplace as documented in popular business and marketing literature. In bringing these ideas and experiences to this volume, I hope to introduce the process of HCI as part of a broader activity: specifically, the development and creation of user experience in a consumer economy.

THE BOUNDARIES OF USER EXPERIENCE

The definition of user experience (UX) and its relationship to HCI is complex. Both fields share boundaries with a number of other fields, and each other. On one hand, either field can resemble anthropology, cognitive psychology, industrial design, or computer science in practice. On the other, customer relationship management and marketing play a large role in actual day-to-day experiences with products and services. Consulting for a broad range of organizations on projects ranging from consumer products for broad audiences to highly focused products for internal use has shaped thinking about the definition of the term. User experience is a set of broader considerations than HCI. It aggregates and contextualizes HCI by incorporating the concerns of both end users and organizations. In other words, the user experience consists of all of the factors that influence the relationship between the end user and an organization, especially when a product[1] mediates that relationship.

UX Is Context

From the users' perspective, their experiences are continuous. The products, their immediate environments, and their lives all interact and feed back on one another. On the most basic level, what someone understands about a product affects what he or she finds attractive about the product, and what is attractive affects his or her willingness to understand it. How much depends on the rest of the context, but it is a mistake to think that only the look or the functionality matter. It all matters, and research and iterative design determine to what degree.

TABLE 46.1. Organizations Involved in End-User Relationships

Product	Organizations Involved
Traditional technology product	Sales/Repair
Traditional desktop software	Sales, Support
Website	Internet Service Provider, Website Provider
Mobile	Handset manufacturer, Network provider, Application provider, Content provider

Many seemingly stand-alone products now are merely ways to access services provided by organizations. End users' relationships to an experience and the organizations creating the experience intertwine more than ever (see Table 46.1). In the days of traditional industrial manufacturing (roughly before 1970), end users of a product may have only had one interaction with an organization: the store from which they bought it, which may have also provided support and repair services. Packaged software included three or more: the store that sold the hardware, the store that sold the software, and the providers of technical support. With the introduction of web-based software interactions, the number of organizations increased, with the addition of an ISP and website provider. Modern mobile phone based applications may involve even more: a handset manufacturer, an operating system developer, a network provider, an application developer, and a content provider. All of these organizations contribute to the end-user experience, often without a lot of coordination between them.

HCI is part of a technology creation process. Like any technology creation process, doing it right requires not only automating a certain set of tasks, but also inventing tools that introduce new possibilities for both the people using them and the organizations creating them. In such a multilayered environment, product development can go in many directions, and research can be conducted almost ad infinitum. However, in the end, limited resources require choosing one promising direction.

User experience design and research is a pragmatic pursuit. Its goal should be the understanding of the experience of technology users and technology-producing organizations to manage the risks of technology creation and increase the chances of success.

Garrett's Elements

Garrett (2000, 2002) developed a model (see Fig. 46.1) for understanding how various aspects of product design interact to create a whole user experience.

Garrett (2000, 2002) focused on web design, but his model extended to most other kinds of user experience. It described

[1]I define *product* broadly. A product represents the interface between an organization and end users. It could be a physical object, a service, a system, software, or a combination of these. For example, an ATM consists of three elements: the machine itself, the card used to access it, and the service that it enables access to. However, it is a single product, especially from the perspective of the end user. More often, it is a single definable entity, but seemingly, stand-alone artifacts regularly turn out to belong to a system of interlocking, interdependent elements.

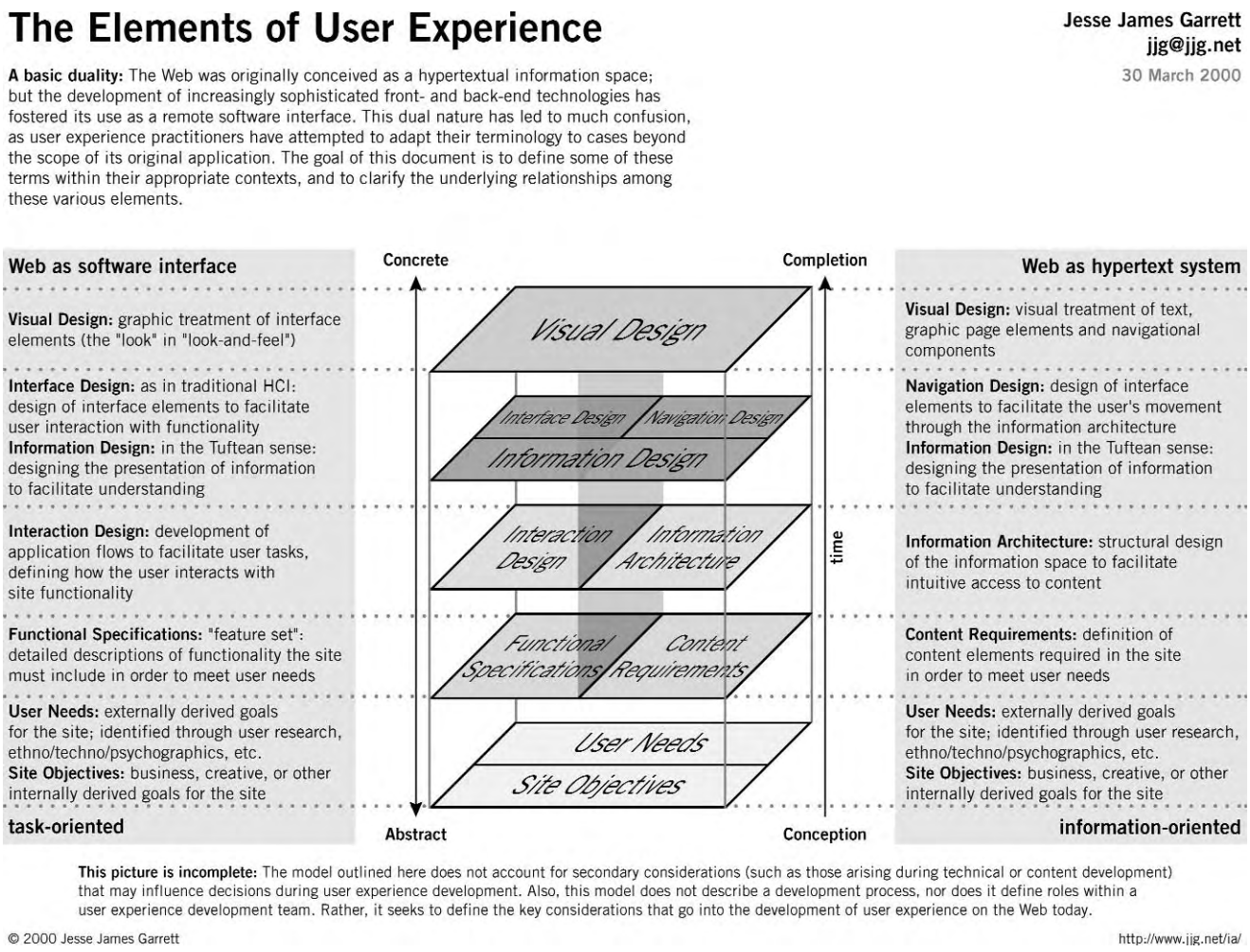


FIGURE 46.1. Garrett's elements of user experience diagram (Garrett, 2000).

the dependencies connecting abstract business and user goals to visual design through a set of intermediate steps. These steps describe the information that a product provides and describe how people interact with that information. Productivity products (the left-hand column, defined as "web as software interface") emphasize the content less than the interaction, while information products ("web as hypertext system") emphasize the content more than the interaction.

The diagram defines stages in understanding and managing this process, and emphasizes that factors that are unrelated to ergonomics or functionality constrain end-user experience. It implies that good HCI is a subset of good product development, and inseparable from the larger context. The outer layers in Garrett's diagram hide the inner ones from both users and from the organization at large. Users only see the visual design layer, while organizations only see the website objectives layer.

However, the user experience depends on a cascading sequence of assumptions and decisions. These are constrained by economic factors imposed by the organization and psychological or sociological factors imposed by users and society. These economic, psychological, and sociological factors tell at least half of the story of the complete user experience. They define the context in which decisions are made and the product actually experienced, and they should be the ones in which it is designed.

THE ORGANIZATIONAL EXPERIENCE

End users are not the only customers of a given piece of technology. Technology creation solves two sets of problems: one for the people using it, and another for the organization creating it. HCI research and design often assumes that an organization's goal is to provide optimal end-user experiences, but many other factors drive organizational motivations. Organizations' needs and desires[2] frame and prioritize product research and

[2]Hassenzahl (2003) used *pragmatic* and *hedonic* product attributes to discuss roughly these same concepts. His terms refer to individuals' perspectives in the abstract, but I prefer to use *needs* and *desires* because these terms better frame discussions from users' perspectives and work better when discussing the parallel between an organization's perspective and that of the user of its product.

development as much as users' abilities and goals, which are the traditional realm of HCI.

An organization creates a product because it desires something from a user base. The difficulty is that the user base often desires something different. The resolution of these two disjunctive desires deeply affects the final user experience. For this reason, user-experience design and research starts with organizational strategy.

This example from industrial design foreshadows many of today's HCI and user experience issues 80 years earlier.

The 1927 Ford Model T

The Ford Model T was an incredibly successful car, and the first killer app of the 20th century. Throughout the 19 years it was manufactured, its design remained unchanged, except for one thing: Every year, it was cheaper than the year before. From the perspective of Model T users, it was a great vehicle: reliable, predictable, and inexpensive. However, by the mid-1920s, it was not selling well relative to many of its competitors, and Ford discontinued it in 1927 (Wikipedia, 2005a). Henry Ford refused to value anything but efficiency (for his company and its customers) in his products. However, by the mid-1920s Ford's competitors were selling more cars by evolving the look and feel every year (*styling* in automotive terminology). The goal went beyond making cars more efficient or cheaper to making them look different. Having realized that people treated cars as expressions of identity, the competitors included styling as a key part of the user experience.

Ford had many options they could have pursued in response to the economic pressure put on them by the profits lost to competitors. They could have restructured their manufacturing processes to make Model Ts even cheaper. They could have lowered the quality of their product to increase their margins; they could have embarked on a research and development program to merge their car, tractor, and airplane products, so they would only produce one product. They could have laid-off workers and decreased the number of cars they were producing and so on. Each plan would have differently affected the driver experience. Ford's decision was to stop making the Model T and introduce the 1928 Model A, a car with competitive styling (available in four colors, none of them the black of the Model T; Wikipedia, 2005b). Ford's industrial designers then updated the styling of their cars on a regular basis as their competitors did.

Beginning user experience evaluation by analyzing the sponsoring organization's motivations regularly reveals the issues that pervade the assumptions behind the product. Introducing subtle changes in core assumptions, as Henry Ford's son Edsel (then the President of Ford) did in 1927, can change the experience of the entire product without having to rethink the whole user interface (because the problem may not be in the interface at all).

A Children's Art Product Manufacturer Website

A maker of children's art products wants a new information architecture for their website. The website has three audiences (children, educators, and parents and grandparents), and more than 200 different kinds of content. With such a depth of information and such a broad audience, there is no obviously canonical way to structure the content. The historical function of the website as a sales channel directed toward parents and educators guided all of the initial architecture choices. However, interviews with company executives responsible for the website revealed that these assumptions were either inaccurate or inappropriately emphasized. Most mistaken was the belief that the site had to be a revenue source. In fact, the Chief Financial Officer (CFO) flatly stated that the website's goal is to spread the company's brand identity as broadly as possible among their primary audience. In its incarnation, the website met neither the goals of the original development team nor its actual goal as a brand vehicle.

Throughout the product's development lifecycle, internal expectations and assumptions guide the experience it creates in subtle ways. In this example, the information architecture for a website was distorted by the explicitly stated goal of revenue production, even though the organization's leaders had changed their goals. When expectations contain internal conflicts, they produce contradictory and confusing interaction.

Organizations have to put themselves first, even when creating products for end users. For example, Southwest Airlines policy allows customers to apply the price of an unused ticket to another ticket. However, the company profits if people do not take them up on the offer. Thus, it is not in the best interest of the company to make it easy to perform the transaction. As of October 2005, Southwest.com allows the user to transfer funds from an unused ticket only if they have the exact confirmation number of the unused ticket and the exact spelling of the name associated with it—even if they have an account on the Southwest website and the system database can pull up all of the other account information. The Web-site interface makes transferring funds difficult because the interface ultimately serves the company's financial interests, not those of the customer.

User experience defines the boundaries of product development through stakeholder needs and end-user goals. These needs and goals are not just management requests or customer complaints. They represent the core of how the organization defines success and what end users expect the product will do for them.

Applying the tools of user experience research and design to the organization is tricky. Looking closely at organizational assumptions and expectations steps right into in-house politics—that aspect of collaborative work that everyone would prefer did not exist—and can create interpersonal tension. However, unstated internal priorities often inhibit successful user experience design more than any external factor, so they are important to investigate. Fixing office politics is outside this chapter's scope and most readers' job descriptions, but explicitly clarifying an organization's priorities is well within the capability of an HCI professional. In fact, it is critical. As we have seen, confusing, conflicting, and ambiguous organizational agendas produce conflicting product requirements, which in turn produce difficult to use interfaces. Knowing organizational needs helps balance the needs of users and organizations in design.

THE USER VIEW

As stated above, factors that affect an end user's experience are not just those that determine the efficiency of the interface in enabling task completion. Functionality is, of course, critical to the continued product viability—it needs actually to do something—but viability is more than functionality. We all willingly enter into experiences (buy products, use services, etc.) that are far from functionally optimal, and yet we leave satisfied. Agarwal and Karahanna (2000) defined the concept of "cognitive absorption," which seems like a good way to describe the main goal of product designers and developers, as "A state of deep involvement . . . exhibited through temporal dissociation, focused immersion, heightened enjoyment, control, and curiosity."

Few products regularly produce cognitive absorption. In order to understand why, it is valuable to define some other terms describing important aspects of the user experience from the user's perspective.

Ortony, Norman, and Revelle (2005, p. 174) proposed a model that describes "an organism's" (e.g., a person's) psychological function in the world. The model's four (continually interrelating) parts are:

- Affect—what the organism feels
- Motivation—what the organism needs and wants
- Cognition—what the organism knows, thinks, and believes
- Behavior—what the organism does

Product design implicitly takes all of these factors[3] into consideration, but explicit examination of them is rare. Marketing researchers investigate motivation; interaction designers use their knowledge of cognition; usability research focuses on behavior; and visual or identity designers and advertising agencies try to influence motivation through affect. However, that is an ideal situation. In reality, the practice of understanding and structuring a unified experience is so new that design generally runs on gut-level intuition, and everyone is guessing at everything. Gut-level decision making is not necessarily bad. Humans are often good at predicting other humans' reactions—except when intuition totally fails.

The User Experience of Products

Affect. According to Ortony et al. (2005), emotional response, or affect, is a complex interaction of immediate reactions modulated by experience with previous situations and cognitive prediction of future states, all of which happens rapidly and simultaneously. Immediate feelings, emotions, and moods are different states operating at different levels of granularity. They are also critical to people's experience of a product. When people fall in love at first sight with a product or a place, their successive experiences will not be moderate. The emotions may lead them to overlook interaction problems or poor functionality. Later, the emotional state may wear off, the honeymoon ends, and the inadequacies of the product turn joy into disillusionment.

Davis (1989, p. 320) showed that "both *perceived usefulness* and *perceived ease of use* were significantly correlated with self reported indicants of system use." People's emotional relationships to products before they had the opportunities to use them affected how they used them later. Zhang and Li (2005, p. 106) extended Davis' research by applying a more primal concept, affective quality. They investigated the perceived affective quality of software products and concluded, "a user's immediate and reflexive affective reaction to [information technology] has a positive impact on his or her consequent cognition-oriented evaluations of the [technology]."

Furthermore, Nass and Reeves (1996) described in detail how people exhibited many of the same emotional responses to computers, televisions, and films as they did to other humans, significantly changing their expectations and behaviors toward the technology as a result.

What constitutes affective quality (which is measured in terms of valence and activation; e.g., the direction and magnitude of the emotional response) in terms of technological products is still under investigation. However, evaluating and designing the complete user experience clearly requires close consideration of the experience's affective aspects.

Value. People act for a reason. They engage with products or experiences for some reason (or reasons), they keep using them for other reasons, and they stop for others still. In the largest context, Maslow's (1943) hierarchy of needs serves as one model of how what people value in their lives motivates their actions. Norman et al. (2005) described how one kind of motivation, curiosity, could arise from an emotional response to an environment. "Animals' motivation systems [let] the resting point of affect be slightly positive so that when there is nothing that needs to be done, the animal is led to explore the environment" (Ortony et al., 2005, p. 194).

However, pure curiosity rarely leads people to have new experiences or to continue well-known experiences. When using a household appliance, for example, curiosity rarely drives people's behavior. From a product developer's perspective, a good approximation of motivation is what creates value for the end user. Value consists of two elements:

- The product's perceived potential for changing a customer/user's life
- How well it satisfies that potential

Perceived potential consists of three elements: functional, economic, and psychological (Sawhney, 2003). The functional aspect is the prospective user's expectation about whether the product will be able to solve a real-world problem the person

[3]These terms are a framework for the subsequent discussion. They are not defined in the same rigorous, technical way that Norman et al. (2005) defined them in their work. The definitions provided here are the more common dictionary definitions, which are a superset of how Norman et al. defined them.

is having. "Will the disk utility program recover my thesis?" or "Will the personal video recorder let me watch *The Simpsons* at 3 A.M.?"

The economic aspect consists of the cost-benefit analysis that a prospective buyer of a product does when considering whether purchasing the product will be worth the opportunity cost of spending money on it. This is the literal, most traditional, definition of *value*. "Will this CRM system let me shave 25% off of my expenses?"

The psychological aspect contains all of the hopes that someone has for how owning or using a thing will change his or her life and is the both the most difficult to understand and the potentially most important. It holds all of the emotional attachment, all of the social pressure, and all of the personal desires that make up someone's self-image, as they are contemplating buying, and then using, a product. Some consumer objects, such as the Nokia 7280 phone (see Fig. 46.2), evoke much more about their values than they communicate about their functionalities. Designed as fashion items, much of their functionalities are the same as that for garments: They explicitly project an image of their users to both others and the users.

However, these same ideas apply to ostensibly purely functional products. Every underused enterprise software product is the result of a perceived value that did not match the reality of the situation on the ground, often for reasons that were neither functional nor economic.

The design of the user experience is the practice of creating products that satisfy perceived value.

What brings people value changes with context is that, at different places and times, people will have different values. There is a lifecycle to expectations dictated by habituation. As same people grow accustomed to a product's functionality, its novelty wears off. For a long time, the Model T satisfied what consumers wanted in a car. For 19 years, Henry Ford thought only the price of the car had to change, but consumers clearly thought differently. As the automobile's functionality became commonplace, people's relationship to it changed. They began to focus on the psychological needs it satisfied and to see it less as a tool they were using and more as part of who they were. People desire variety (Postrel, 2003), and the black Model T no longer satisfied. Car buyers were willing to pay extra for a different user experience, but Ford did not recognize this until it was almost too late.



FIGURE 46.2. Nokia 7280 phone.

Blindness to the larger user experience also exists in the development of software products. The business press regularly describes the struggle between well-established companies and their younger competitors. Such stories typically describe a company with an older product whose target audience no longer wants sees value in the user experience their product provides. The older company clearly produced good user value at one point, or else they would not have had the success that allowed them to be in a threatened leadership position. Their products changed their audience's expectations, but then the company failed to notice when expectations moved on. For example, Yahoo! search technology was lagging in the early 2000s when compared to Google's. At one point, Yahoo! was a dominant player in the search market, but by 2005, they got to the point where, "The company is doing everything that the fertile imaginations of their software engineers can muster in order to persuade people to search with them first." (Andelman, 2005).

Likewise, organizations also often produce products for which the market is not yet ready. In 2005, a number of large organizations have invested in entertainment PCs, which look like stereo equipment, and associated products, such as media servers, but, to the puzzlement of the companies making these services, there has been a lack of widespread adoption of such services in the past (Buckman, 2002). These products' unpopularity may have nothing to do with the feature set or its presentation. The makers of these products should not necessarily have been doing any more usability testing or focus groups. The interface for the TiVo (2005) personal video recorder was widely praised by both interaction designers and users, but it took the company eight years to achieve profitability. It may be that patience is an ingredient in the user experience of these products before they appear worthwhile to a broad audience.

As Sawhney (2003) described, the process of creating customer value in technology products requires understanding the interaction of all the elements that make the product desirable:

> According to HP, the benefits of the iPaq are its powerful processor, bright screen, expandability and flexibility—a statement of functional value. But to close a sale, HP must also demonstrate economic value with quantified estimates of improved productivity for end users as well as application developers. And HP must convince customers of the emotional benefits of choosing a device platform that is backed by reputable and financially solid companies such as HP and Microsoft. (Sawhney, 2003)

Creating a user experience requires understanding this entanglement of ideas as well as HP did in creating the iPaq.

The User Experience of Organizations

Brand. Brand identity generally refers to the combination of all the implicit values an organization communicates about itself, as understood by the consumers of that organization's products or services. Symbols such as logos and slogans evoke brand identity, but the actual identity is the set of values that people project onto an organization, and by extension, onto its products based on personal experiences with that organization and its advertising. In terms of the user experience, brand identity creates expectations for the value that an organization's products will provide to the

end user. As such, it is an important component in setting people's expectations for how to approach a product and what the product will do for them economically, functionally, or psychologically.

Brands live in the minds and expectations of the buyers and users of an organization's products and services. A logo can evoke a set of feelings and expectations for the value that a product will give someone, but it is not the actual value. The product still has to provide the value, although often that value is not in terms of the actual functionality, but rather in the emotional satisfaction that owning, using, or being seen with a product brings. This aspirational component of a brand is the emotional value the audience perceives the product will deliver. In that sense, it is the perceived affective quality of all of the products produced by an organization.

Products that do not meet brand expectations can either disappoint or confuse users. During the dot-com boom of the late 1990s, many companies attempted business models that took their brands well outside of people's existing expectations for them.

For example, when Intel, a chipmaker, partnered with toy manufacturer Mattel, it seemed like a good way to merge cutting edge technology with toy manufacturing. The partnership produced several products under the Intel Play brand (Fig. 46.4). However, sales of the toys did not meet expectations, and the partnership was dissolved. As with any enterprise, the circumstances were complex, but one of the potential problems may have been that the Intel brand strongly connoted an entirely different set of values than was appropriate for the sale of toys. As manufactured and sold by Digital Blue (Fig. 46.3), an educational toy company founded to market and develop the products from the failed venture, the products developed by Intel Play are seeing financial success. This shows that the entire hierarchy of Garrett's (2000) Elements can be satisfied on a functional level, but if the total user experience does not fulfill the user's larger expectations, products can still fail.

Good experiences while using a product will affect people's perceptions of the organization that produced it, which in turn affects their expectations for the functionality of other products that the company produces. Bad experiences with a service, such as documented in Rafaeli and Vilnai-Yavetz (2004), can lead to a wholesale dissatisfaction with other products that the organization produces, irrespective of those products' immediate user experience.

From an HCI perspective, understanding and incorporating brand identity into the experience is important. As Saffer (2002) said, "Navigation, nomenclature, and content presentation must also reflect the company's brand. The most elegant visual design in the world isn't going to overcome inappropriate interaction design."

For example, knowing the children's art product manufacturer (previously mentioned) was more interested in communicating the company brand than producing revenue changed the direction of the user experience dramatically. websites intended to efficiently sell products that are designed to be purely functional, whereas one intended to evoke a sense of playfulness, whimsy, and creativity (the psychological values the company in question tried to communicate) is much different. Compare the McMaster-Carr website, which has been a very successful sales website (Spool, 2005), to the site for the Lego toy company (see Fig. 46.5 and Fig. 46.6).

The interaction design, the organization of the content, the kind of content presented, and the visual design of individual interface elements of the two websites differ not just because the audience differs or the products differ (though those differences are undeniably important) but also because the message they want to communicate differs. Compare the Carhartt clothing company's websites in the United States to that in Europe (see Fig. 46.7 and Fig. 46.8). In the United States, Carhartt is branded primarily as a work wear manufacturer, while in Europe, it is a fashion brand for urban youth.



FIGURE 46.3. The Digital Blue Digital Movie Creator, II.



FIGURE 46.4. The Intel Ray Digital Movie Creator.

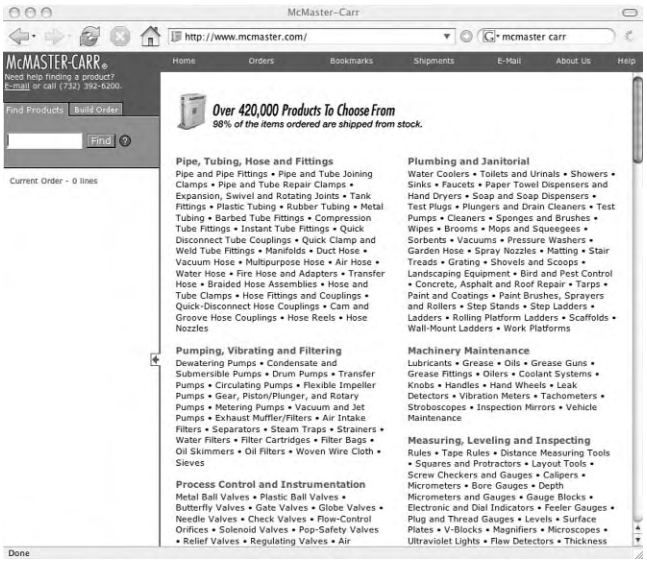


FIGURE 46.5. McMaster-Carr website homepages.



FIGURE 46.6. Lego website hompages.

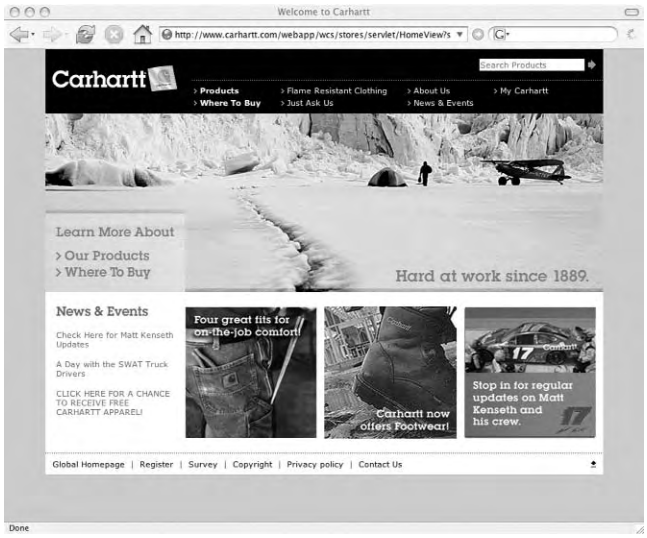


FIGURE 46.7. Carhartt U.S. website homepages.



FIGURE 46.8. Carhartt Europe website homepages.

Relationships. In today's world, we rarely interact with an organization only once. The process of buying, owning, using, and maintaining a product, whether software or an appliance, consists of many interactions with an organization. Customer relationship management (Wikipedia, 2005c) and customer experience management (Wikipedia, 2005d) practices define these interactions as contact points or touch points (Schmitt, 2003). These practices aim to analyze and design positive experiences during these interactions. In fact, some theories (Pine & Gilmore, 1999) claim these interactions are even more important than the products that spark them.

The mobile phone is an example of the numerous customer relationships involved in owning and using a contemporary product. Although technically a computer, a mobile phone is not just a computational tool. Its functionality as a tool and as a communication medium completely depends on the services accessible through a handset. In a sense, it is the physical manifestation of a set of virtual, continually shifting services (as evidenced by the complexity of subscription plans). Without the services, a phone handset is useless. However, the network does not just provide transparent connectivity; the ecology of organizations involved in delivering the mobile user experience is fragmented (Fig. 46.9), and none of the players is wholly responsible for the HCI.

Mobile user-experience design processes require an understanding of the relationship between various organizations and the way in which users will interact with them. Knowing these contact points can focus and prioritize the HCI research and design. Arnall (2001) cited constraints imposed network per-

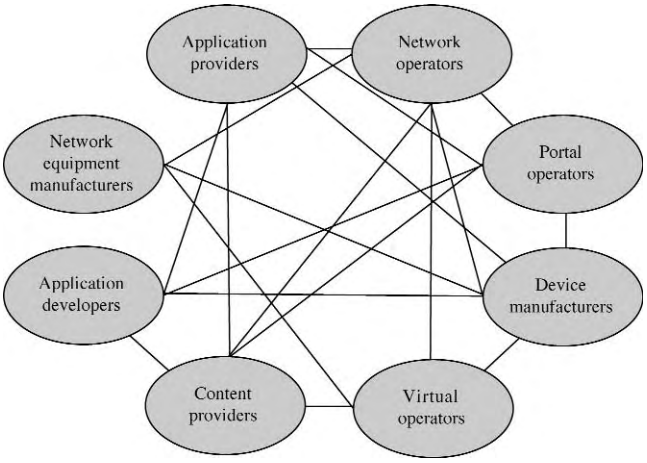


FIGURE 46.9. The mobile data value web (European Information Technology Observatory, 2002).

formance, billing, and hardware limitations in creating an SMS-based service. Creating a satisfying user experience required determining both user's experience with each of those contact points and the integration of all of them. For example, the design of the service had to include both interaction and financial incentives for people to sign up for the service (the signup process was made to be quick and the service was initially free).

The exact nature of contact points will vary based on the details of the service or product under consideration, but it typically involves

- Customer service
- Billing
- Sales
- Account management
- Marketing

To some extent, this has always been true in all HCI development, but it has not been a prime focus of the research and design process. In an ecology of many interacting services, such as described above, ignoring the other players in the environment is no longer optional. When such a service provides a solution to an end user, the solution cannot just be evaluated through the completion of a narrow set of tasks. It needs to be analyzed in the improvement it makes in the life of the person who uses it. People must find value throughout their interaction with it, whether through the out of box experience (Wikipedia, 2005e) in unpacking the product, or how they feel as they are using it, or their interactions with the product and the organization during a technical support call.

Industrial designers and architects have addressed these issues for a long time, recognizing that the evolving roles their products play in people's lives are not always possible to predict or design to the last detail. They have focused on creating user experiences that offer multiple channels of value (rarely in monetary terms, but by a combination of affective and functional ideas). Salespeople and marketers have approached the experience from the other direction. They try to identify the interactions people have with an organization, to understand the value (in monetary terms) of those interactions, and to maximize their monetary values or to minimize their expenses.

Computer interfaces straddle both sides of the equation, providing immediate value for end-users and—especially in a dynamic networked environment such as that provided by mobile phones, ubiquitous computing, or the web—value for organizations (whether monetary or, as in case of governments or nonprofit organizations, other through other metrics that include social goods). Integrating an analysis of the relationship between people and organizations as mediated by the interface is a key component to providing value to both groups.

EXAMINING THE USER EXPERIENCE

Approaching the investigation of such difficult to quantify ideas as affect and value is no small task. Organizations may be unable to articulate their intentions or values. Differentiating end users' needs from their desires and their actual behavior from hopeful visions is difficult. Further, the ambiguous nature of the collected data makes interpretations vary across interpreters. Extracting quantitative information about a broad group of people takes an investment of extraordinary resources.

However, the difficulty of collecting this information should not discourage you from trying to collect it. In order to reduce the risk of failure (though, sadly, probably not increase the risk of success), a model of the whole user experience such as Garrett's (2000) is valuable.

This section describes in detail several techniques for understanding the organizational and user needs for the user experience. They are by no means exhaustive, but they are included as examples of how to approach a user-experience research project, rather than focusing on fragmented tasks, and how to pragmatically apply the theory of the previous sections.

Identifying Organizational Goals

There are three steps to understanding organizational goals for a product:

1. Identifying stakeholders
2. Collecting stakeholder goals
3. Prioritizing among the goals

Identify stakeholders. Start by identifying groups who most often own the product (or who most often care about the product). Make a list of all of the departments who affected by the product's success or failure and of who in each department is most responsible for it. If there is not a single person who is responsible for the product in a given department, find the person who dealt with it most recently. Odds are that this person regularly deals with it or can tell you who does. Product managers generally know which groups and

individuals have the biggest stake in the project and the list will likely contain:

- Engineering[4]
- Design
- Marketing

Other groups can have stakes in the process, depending on the size/structure of the organization in the product's success. A significant managerial presence in a product could be a major moneymaker (or loser) or if it is brand new. Each of these groups has a different perspective on the product.

For example, a fictitious list of stakeholders (Table 46.2) for a web-based data warehousing application contains representatives from identity design and marketing in addition to the people who actually build the product.

Collect stakeholder goals. Once you have your list of stakeholders, find out what they consider the most important issues. You can do this either by getting all of the stakeholders together and spending an afternoon setting organization-wide priorities for the product or by speaking to each person independently. Individual interviews are often necessary with executives, and it is critical that they are involved in this process. Ask each person (or department)

1. In terms of what you do on a day-to-day basis, what are the goals of this product?
2. Are there ways that it is not meeting those goals? If so, what are they?
3. Are there questions you want to have answered about it? If so, what are they?

Every group will have different goals and will measure success differently. Programmers may measure success by the number of bugs per thousand lines of code. Identity design may have internal reviews that evaluate how well the product integrates with the corporate brand. Customer support will want to minimize the number of questions they have to field. Sales will always want to bring in more revenue.

Once you have spoken to the departmental representatives, make a list of the goals and desires. At this point, you will probably see that some of the goals are contradictory. It is too early to attempt to resolve the contradictions, but investigating the relationship between them may be an important near-term goal for the project.

Prioritize organizational goals. Based on your interviews, you will have some idea of the corporate priorities with respect to the goals you have defined. Some things are important because the organization believes they prevent people from using a key feature. Others may be important because they differentiate the product from its competitors. Still others might be less important because they create a drain on resources or are currently a topic of debate within the company.

TABLE 46.2. List of Potential Stakeholders of a Fictitious Data Warehousing Application

Alison, VP of Product Development
Erik, Interaction Design
Michel, Marketing
Claire, Database Administration
Ed, Customer Support
Leif, QA
Joan, Identity Design

TABLE 46.3. A List of Goals and Questions of a Fictitious Data Warehousing Application

Who	Goals and Questions
Alison, VP Product Development	Fewer complaints from major clients Match data retrieval features offered by competitor
Erik, Interaction Design	Help construct more sophisticated reports, since the current interface does not reveal full report engine Why do so many people start and then abandon the query wizard?
Michel, Marketing	To show tight integration of the new report generator with the query system
Claire, Database Administration	Is there a way to keep people from clicking the search all button? It hammers the database every time.
Ed, Customer Support	Reduce support calls about report generator Shift more support from the phone to email
Leif, QA	Identify query wizard JavaScript errors to address user complaints
Joan, Identity Design	Make the look and feel of the acquired report generator match that of the query interface

There are many prioritization methods. Sometimes, just making a list is sufficient, but using a technique that abstracts key factors can be useful. Table 46.4 explains one modified from the total quality management industrial manufacturing discipline.

Using this technique, the questions in Table 46.3 could be prioritized, as in Table 46.5.

Often, when prioritized systematically, it is easy to see why product development happens in the way it does. The lists show unstated company priorities come out and agendas that are orthogonal to the organization's actual needs. In retrospect, it is possible to see how decisions go against the product and organization's needs and how teams' abilities produce the conditions that generate bad user experiences. Most importantly, tables such as these allow you to prioritize what you learn about user needs.

A rapid technique: project history. It is not always possible to perform a rigorous investigation of an organization's

[4]These terms are used here broadly. *Engineering* typically consists of programmers in a software or web environment but can include electrical and mechanical engineers in a hardware-development project. Likewise, *design* can include information architects, industrial designers, interaction designers, and visual designers.

TABLE 46.4. A Prioritization Technique

1. Make a column next to your list of questions and label it "Desire." Go down the list and negotiate with the group a rating for each item on a scale of one to five. Five means the feature affected is a must have, critical to the success of the product, and one means it is a nice to have, but not essential.
2. Next, make a second column and label it "Risk." This will reflect how bad the problem is. Write a number on a one to five scale here, too. Five represents bad problems (ones that either directly affect the bottom line right now or represent major malfunctions), and one refers to problems that are annoyances or information that would be good to know.
3. Finally, make a column and label it "Ease." This is how easy your team feels it will be to address the problem. Five means that it is easy to do, and one means that it is very difficult.
4. Multiply the three entries in the columns, and write the result next to them in a third column called "Priority." This combines and amplifies the factors. Ordering the list by the last column gives you a starting order in which to investigate the product's user experience.

TABLE 46.5. The Prioritization Technique from Table 46.4 Applied to the Questions in Table 46.3

Goal	Desire	Risk	Ease	Total
Match data retrieval features offered by competitor	4	3	2	24
Why do so many people start and then abandon the query wizard?	4	5	4	80
To show tight integration of the new report generator with the query system	3	3	4	36
Is there a way to keep people from clicking the search all button? It hammers the database every time.	5	5	3	75
Reduce support calls about report generator	2	4	2	16
Identify query wizard JavaScript errors to address user complaints	3	2	5	30
Make the look and feel of the report generator match the query interface	5	2	4	40

needs. A fast way to understand the organization's goals is to create a quick history of the project. The sequence of events that lead to the current situation reveals a set of problems and solutions, which in turn reveal what the organization's needs and values. The process is straightforward in principle, although the answers to basic questions can reveal complexities in priority and interest that a simple narrative explanation of the current situation does not. Getting a project history can be as simple as asking the following questions of the key stakeholders responsible for a project. The goal is to encourage them to describe the sequence that led to the current situation.

- Why did you decide to this?
- Why did you decide to do it now?
- Who initiated the project?
- What was the organizational pressure that suggested it?

The idea is to ask these questions (which are just a variant on the standard who/what/when/why interrogatives) recursively. In other words, for every answer, it is possible to ask the same questions to get an even older, and maybe deeper, set of motivations. Some techniques recommend doing a certain number of times (four seems to be common), but going deeply on a couple of key ideas is usually enough to understand the deeper motivations and constraints underlying the current situation. One variant that has proved useful is to ask to include anyone mentioned into the conversation. "Oh, so Lucie suggested that PCB designers weren't using the spec sheets, which is why we are trying to make them more prominent. Could we talk to her about how she determined that they were not using them enough?" It could be that Lucie has stacks of e-mails from customer service in which people ask for information that is readily available, or maybe she just has a hunch. In the former case, the information in the e-mail could be valuable in determining users' expectations from the service; in the latter case, understanding Lucie's motivations provides information about how she measures success or envisions the purpose of the service.

Field Observation

Norman (1998) said, "The goal is to make the people who are being observed become participants in the discovery process of learning just what their real needs are—not the artificial needs proscribed by the way they do things today, but what the goals are, what they are striving for. This is the role of rapid ethnography."

A highly effective and increasingly popular method of exploring the user experience comes from field-research techniques based on methods pioneered by anthropology, ethnography, and ethnomethodology. Examining work and life context produces a richer understanding of the relationships between preference, behavior, problems, and values. Laboratory and survey methods extract people from their environments to focus on individual tasks or perspectives or aggregate responses from many people. Field observation's goal is to gain insight into the total relationship between the elements of the user experience as experienced and understood in the context of use.

Rather than trying to validate theories in a controlled setting, these ethnography-derived methods, including contextual inquiry (Beyer & Holzblatt, 1998), derive insight through direct observation of people in their actual environment with (ideally) little presumption about their behavior and needs.

Direct observation removes much of the bias that creeps into research when people or tasks are isolated. Outside of the environment that triggers them, our explanations of desires, values, reactions and behaviors, especially in routine events, lose critical details by our tendency to simplify, idealize, and project. Exploring the context of activities can identify people's larger goals through the small details. For example, when someone leaves a note on a kitchen counter, the goal is not to just to leave the message, but rather to communicate something specific to a member of the household (even him- or herself). The message may be a to-do list, a reminder, or an alert (Elliot, Neustaedter, & Greenberg, 2005), and its location communicates how to interpret the message. When discussing domestic communications

outside the context of their daily routine, critical details such as spatial placement, time of day, materials used, or triggering event can be lost.

Direct observation identifies emotional reactions that would be otherwise difficult to capture. For example, Vrendenburg, Righi, and Isensee (2001) described a situation where a t-shirt included in the packing material of an IBM RS/6000 computer led to surprise and delight from users—signs of a good user experience—just unpacking the box:

> Users opened the product box to find a t-shirt, a mouse pad, a copy of Wired magazine, and games that showcased the 3D graphics capabilities of the system such as Quake. This approach to design worked beautifully. It became cool to have an RS/6000. One of the most common questions asked by customers in the feedback survey was "Where can I get another t-shirt"? (p. 34)

This was an unexpected observation that was not part of a focused program of focused ethnographic observation of people's experiences unpacking RS/6000 computers, but it is representative of the kinds of things such observation produces. In another instance, Berg, Taylor, and Harper (2003) observed the following relationships between UK teenagers and their mobile phones:

> [The] text messages that were exchanged were sometimes described as objects that evoked particular memories. The messages were the embodiment of something personal that could be stored, retrieved, reread and shared, becoming tangible mementos for individuals and groups. Thus, the phone appeared to provide a means to participate in social exchange in so far as it enabled particular objects to take on symbolic meaning and for the objects to be seen as meaningful between people. (p. 434)

Such insights map directly to user experience design (as the authors then proceeded to do). They allow technology to enable specific, observed behaviors in the context they occur, rather than hypothetical behaviors and assumed needs.

Field research methods for user-experience design are typically neither as detailed, data-heavy, or analytically rigorous as formal ethnography (Bentley et al., 1992). These techniques focus on pragmatic on the ground observation and interpretation within the context of a development and production process. They use standardized methods and seeking to identify contact points, activity sequences, artifacts, and values in the context of work practices. Beyer and Holzblatt's (1998) contextual inquiry is probably the most prevalent of these techniques. Generalized from rapid ethnography (Millen, 2000), Table 46.6 lists a set of steps for conducting field research.

TABLE 46.6. Steps for Conducting Field Research, Adapted from Millen (2000)

1. Find key informants
2. Narrow the focus
3. Use interactive observation
4. Use multiple researchers and analyze collaboratively
5. Validate conclusions

Find key informants, schedule research. Millen's (2000) recommended identifying informants and asking them to serve as guides through a field observation. He suggested that guides should be "people with access to a broad range of people and activities and be able to discuss in advance where interesting behaviors are most likely to be observed or where activities that reveal social tension are most likely to be found" (Millen, 2000, p. 282). For example, when observing technology in a hospital, it pays to talk to a nurse who works there, or if investigating hobbyist PC case modification (casemod) culture, it's valuable to have a member of a club of modders introduce you to the hobby and the players in it.

When choosing informants, you should pick at least five people or groups who resemble the people who will use your product or who will provide key insights. Overall, they should have the same profile as the eventual target audience, though fringe members of a group may be good informants and provide information or exhibit behavior that typical group members will have internalized.

The breadth and depth of research will determine the extent of the study undertaken: long-term planning generally requires deeper insight and, thus, more and longer observation than short-term problem solving, for example. A typical month-long research schedule (Table 46.7) generally involves two to five hours per observation or interview period, followed by two to three hours of group analysis per hour of observation.

Narrow the focus. The goal of traditional ethnographies is to understand as much as possible about the entire context in which a group of individuals acts, without judgment. In contrast, most commercial research projects begin with an idea about what problems need solving and an idea about how to solve them. Field observation clarifies and focuses these ideas by discovering the situations in which these problems occur and how people deal with them. In addition, unlike an evaluative technique such as usability testing, it is observational and typically uncovers unexpected directions. Thus, it is best done before the process of creating solutions has begun when there is still time to iterate on research. This is usually at the beginning of the development cycle.

However, in the interest of maximizing immediate results, the project typically concentrates on the fields of activity that will likely produce results that designers can incorporate into the user experience. Narrowing focus means identifying the important aspects of your audience's work or life practice, while leaving open the option to challenge assumptions. One technique is for researchers to familiarize themselves closely with the terminology, tools, and techniques their audiences are likely

TABLE 46.7. Typical Field Research Schedule

Timing	Activity
$t - 2$ weeks	Organize and schedule participants.
T	Begin observation. Begin analysis-scheduling process for development team.
$t + 1$ week	Complete observation. Review videotapes and notes. Complete analysis scheduling.
$t + 2$ weeks	Begin analysis.
$t + 3$ weeks	Complete analysis.

to use. An informant can walk the researchers through some concepts before formalizing the research goals. The sportscaster method, where one informant explains what another one is doing, is another useful technique. For example, walking through a shopping district with a fashion-conscious teenage commentator can reveal a lot about where to look for interesting behaviors, rather than starting from scratch.

With this information in mind, it's possible to narrowly define the aspect of the practice that you can ask questions about and observe.

User interactive observation. This is the key to the technique, and it requires going to where people are engaged in the kind of activity the experience for which you are designing and asking them to teach you about their activities. Most of the time should be spent observing what the participants are doing, what tools they are using, and how they are using them. One effective technique is to take on the role of an apprentice and asking them to give a running description of what they are doing. As in an expert-apprentice relationship, this should be enough to describe the practice to the apprentice but not enough to interrupt the flow of the work. As an apprentice, you may occasionally ask for explanations, clarifications, or walkthroughs of actions, but do not let it drive the discussion.

Observations can be in the form of structured interviews, with prewritten discussion guides. This is useful in answering specific questions, but risks missing key challenges to assumptions. Other kinds of tools can elicit specific kinds of information (Beyer & Holzlbatt, 1998; Millen, 2000), or aid in the constructing models later (Wixon et al., 2002). An informant can use a paper model of a shop floor, for example to describe activity in a factory than would be possible in the loud environment of the factory itself.

Collect as much documentation of the practice as possible. Digital and video cameras, liberally used, provide both material for analysis and illustrations for presentation. Collect physical artifacts, when possible. For example, a group of researchers studying patterns of technology use in urban German areas took 400 photographs in a span of three hours and brought back posters, local handicrafts, and a pipe from a construction site.

Use multiple researchers and analyze collaboratively. Collecting and analyzing data simultaneously can provide efficiency, though it introduces more potential biases to the interpretation of the observations (Madison, 2005). Techniques for group qualitative data analysis range from traditional transcript coding methods (U.S. General Accounting Office, 1996) to contextual inquiry's formal methods (Beyer & Holzblatt, 1998) for constructing multifaceted models of users work practices. Affinity diagrams are a particularly popular method. Table 46.8 describes the steps in the construction of an affinity diagram. It takes about one day.

This rather mechanistic process yields good first-cut results about the breadth of the user experience, and frames subsequent investigation.

TABLE 46.8. Affinity Diagram Construction, Adapted from Beyer and Holzblatt (1998)

1. Extract 50–100 notes from each interview. Notes are singular observations about tools, sequences, interactions—anything. Randomize them.
2. Get a group of people together in a room with a blank wall or a big whiteboard. Have them block out the whole day for the work.
3. Divide the group into pairs of analysts. Give each pair an equal number of notes.
4. Write one note on a Post-it and put it on the wall/window/board.
5. Tell the group to put notes that relate to that note around it one at a time. It does not matter how the notes relate as long as the group feels they relate.
6. If no more notes relate to a given note cluster, write a label summarizing and naming the cluster (use a different color so it's easy to identify the labels).
7. Repeat the process with the other notes, labeling groups as they occur.
8. Generally, it is useful to break up groups of more than four notes into smaller clusters. However, there is no upper bound on how many notes may be in a group if there is no obvious way to break it up.
9. As the groups accumulate, Beyer and Holzblatt (1998) recommended using pink notes to label groups of blue notes and green notes to label groups of pink notes.

Validation. A key part of modeling is to evaluate the quality of the model with the people whose lives it models. An immediate follow-up interview with in-depth questions can clarify a lot. Certain situations may not have been appropriate to interrupt (e.g., if you are observing a surgeon or a stock trader, that may apply to the whole observation period), whereas others may have brought up questions that would have interrupted the task flow. Conducting this interview while the participant's memory of the event is still fresh will produce best results. "You'll never understand what's really going on until you've talked to people about what they are doing. The [follow-up] interview . . . gives you the rationale to make sense of things that might otherwise seem odd or insignificant" (Bellotti, 1999).

Focus Groups[5]

People's affective responses and values are hard to observe objectively, and getting a subjective read is often all that is possible. Focus groups are structured group interviews that quickly and inexpensively reveal a target audience's desires, experiences, priorities, and values. Sometimes vilified by their associations with dishonest marketing, they do not deserve their notoriety. They are neither the panacea for curing bad products nor pseudoscientific voodoo to justify irrational decision making. When moderated well, carefully analyzed and appropriately presented, they are an excellent technique for uncovering what people think about a given topic and, especially, how they think about it. A focus group reveals people's perceptions of their values: what they feel right now and how they see that in relation to themselves. Those are crucial in understanding how an experience will affect them.

[5]Much of this chapter is adapted from Kuniavsky, M. (2003).

In product development, focus groups are most useful early in the development cycle, when they generate ideas, prioritize features, and provide insight into people's values and expectations. They can reveal the features people value highest and why they value them, though not whether they will actually use them. As a competitive research tool, they uncover what people value in competitors' products and where those products fail. As Krueger (1988, p. 83) said, "The purpose of focus groups is not to infer, but to understand, not to generalize but to determine a range, not to make statements about the population but to provide insights about how people perceive a situation."

A focus group series is a sequence of tightly moderated group discussions among people taken from a thin slice of a product's target audience. The goal is to encourage the participants to feel comfortable revealing their thoughts and feelings by putting them in a group of people who are like them, or share an interest or an experience that directly relates to a product or an idea.

Prepare. Focus group preparation consists of having several things:

- *A schedule.* The best results come from situations where there has been enough time to examine the contingencies. A good schedule provides sufficient time for everything, especially recruiting and guide writing, and enough slop to make a mistake or two.
- *The target audience*. Who will be to invite to participate? Specifically, you need to know the subset of the target audience that is likely to give you the best feedback.
- *The research scope.* Focus group series can have a few groups of a handful of people or as many as a dozen groups with ten or more participants apiece. The number of groups and people will depend on the complexity of your questions, the depth to which you want to explore the answers, and the certainty with which you want to know these answers. More than four groups per audience are rarely necessary, but two are generally not enough.
- *Specific research topics.* Not all groups feel equally comfortable talking about all subjects and not all subjects lend themselves to group discussion. Carefully chosen topics and a thought-through discussion guide yield the most information without sacrificing the depth of research or the clarity of the results.

Make a schedule. A typical schedule for a focus group series takes about three weeks from beginning to end and should provide sufficient time for recruiting and writing the discussion guide. The process typically takes about three weeks, as detailed in Table 46.9.

TABLE 46.9. Typical Focus Group Research Schedule

Timing	Activity
$t - 2$ weeks	Determine audience and scope, start recruiting immediately
$t - 2$ weeks	Determine broad topics to be investigated, start writing guide
$t - 1$ week	Write first version of discussion guide, discuss exact topic wording with development team, check on recruiting
$t - 3$ days	Write second version of discussion guide with timing, discuss with development team, recruiting should be completed
$t - 2$ days	Complete guide, schedule run through, set up, and check all equipment
$t - 1$ day	Run through in the morning, check times, and adjust guide questions as appropriate
	Do final recruiting check
t	Conduct groups (usually one to three days, depending on scheduling)
	Discuss with observers, collect copies of all notes
$t + 1$ day	Relax. Do something else
$t + 3$ days	Watch all tapes, take notes
$t + 1$ week	Combine notes, write analysis

Pick an audience. From your ideal target audience, you should choose a subset or several subsets that are likely to give you the most useful feedback. The right group will vary from situation to situation. First, you need a solid profile of your target audience, complete with a thorough understanding of their demographic/technological makeup. For example, if you are just looking to find out what existing users value about your service, you want to pick the people who represent the largest subset of your actual audience. If you are looking to find whether a new audience will be interested in what you are developing, a clear specification of who are the potential users will be necessary and what factors will uniquely differentiate them from others. For example, when introducing a new product for use after a car accident, it is hard to get people to predict what they are going to need; however, talking to people who were in car accidents recently may get an evaluation of what could have been useful. A sample profile is in Table 46.10.

The perspective of the members of the subgroups defines similarity. A group of audiophiles will likely be comfortable together regardless of age, whereas 20-year-old and 35-year-old urban restaurant goers probably have perspectives that differ enough to require multiple groups. If you feel that certain groups of people would not feel comfortable with each other, then do not put them together. Income, race, sex, class, age, job, and computer experience can play roles in how people interact in a group situation and how they react to a given user experience.

TABLE 46.10. Sample Audience Profile for Focus Groups Participant Recruiting

Age: 20 to 55
Gender: Separate groups for men and women
Income: Household income over $70,000/year
Computer use: Computer at home or work
Internet use: Internet at home or work. One or more years' experience. Five to ten hours per week for personal use (shopping, reading news, banking, etc.)
Mobile use: Own a mobile phone, used nonvoice mobile services (played a game, SMS, etc.) One or more times in previous six months
Behavior: Were in a noninjury auto accident in the previous 9–12 months, as driver

Develop discussion topics. For an average focus group, you should have three to five main topics to investigate. You should phrase topics in terms of the project as a whole. "Understanding the mental model people use when researching insurance" could be a goal for an insurance brokerage website, while a service that recommended home building contractors could be interested in "Knowing at which point people turn to an external service when doing home repair."

Focus these objectives enough that a group could adequately discuss each one in about 10 minutes. Do not phrase them as questions or issues that other methods (such as a survey) can better answer. A survey could make "A list of our competitors" better than focus groups, whereas "The factors that make Sony's camera phone experience more compelling than ours" is more appropriate.

Write a discussion guide. The discussion guide is a script for the moderator to follow. It creates a consistent framework for the focus group series by asking the same questions in the same order with much the same context. This allows a discussion to bring out the subtleties of the participants' views without shortchanging any of the topics.

Focus group discussion questions should be:

- *Carefully ordered.* Questions get the participants thinking about certain issues and remembering certain events. A careful sequence of questions takes advantage of their frame of mind to make the flow of the group discussion feel more natural, which in turn helps the participants to maintain a creative flow of ideas and produce better insights. In general, questions should flow from the most general to the most specific, with each question narrowing the discussion a bit. There should be planned transitions between topics unless introducing a brand new topic.
- *Nondirected.* Questions should not imply an answer or present a value judgment. They should allow participants to fill in their own thoughts and values. For example, asking, "Which do you think is a better search service, Google or Yahoo?" assumes that the participant feels there are advantages of one over the other. Instead, framed questions neutrally, "Are there any things you like about using the Google search service? Are there things you like about Yahoo? What are they? Can you compare them? How do they compare?"
- *Open-ended.* Avoid constraining answers to fixed responses. Longer, more open responses tell a greater part of the story and tend to be less ambiguous than shorter responses. Rather than phrasing a question in the form "Which of these camera functions are most important to you?" you could ask "Which functions do you use? How often?"
- *Focused on specifics.* Conversely, encourage participants to be specific in their answers. Krueger (1988) recommended breaking down why questions into multiple what questions, explicitly asking for the influences that informed their decision and the attributes of their decision. For example, "How did you decide to go shopping for a new phone plan?" and "What factors went into picking this carrier?" will provide better insight than "Why did you pick Cingular?"
- *Personal.* Out of politeness people attempt tend to generalize their experiences to the public at large or to some hypothetical audience which they are not part of. Since you want to know individual views, values, and experiences, emphasize individual experiences. Formulate question so that they concentrate on people's current and past behavior and opinions, without presenting the option to project. Thus, "If you had to redo your kitchen right now, which of these features would you use to find a home contractor?" is preferable to "Which of these features do you think are useful?"

Granted, fulfilling all of these criteria with all questions is often difficult (writing questions that are simultaneously specific and open-ended is a particularly tricky challenge), but they should be kept in mind as guidelines that should be followed whenever possible.

Analyze results. There are as many ways of analyzing focus group information as there are analysts. Since the information is, by definition, qualitative and contextual, the focus of the analysis will depend on the purpose of the group.

One method consists of the following steps:

- *Quickly capture initial hypotheses.* Immediately after the end of the focus groups, walk through the discussion guide section by section and ask the moderator and observers to recall their thoughts about it: What was unexpected? What was expected but did not happen? What attitudes did people display? What values did they espouse? What interesting statements did they make (and why were they interesting)? What trends did they observe? Which participants provided interesting feedback? What were the problems with the group?
- *Record the groups, and watch the recordings to verify hypotheses.* Merely remembering a situation can miss subtle behaviors. Words are misquoted. Observers fall into groupthink. Reviewing the discussions clarifies ambiguities and reveals shades of meaning.

MANAGE WITH USER EXPERIENCE

When introducing a new technology into a marketplace, there is a risk of failure and of losing money and time on the investment in developing, marketing, and distributing the technology. It is of course possible to create successful technology without having a model of the end user's or organization's needs and desires. However, such successes are essentially the product of lucky accidents. When such success happens, the organization has to identify the elements that led to it. By this point, however, the product's success has permanently changed the market and the organization, and identifying what made it successful is difficult. This supply-first model depends on predictable markets and is passing. In the current environment, "business [needs to be] an adaptive system for responding to unanticipated requests in unpredictable environments" (Haeckel, 1999, p. 10).

Working from the user experience is essentially a demand-first philosophy, continually redefining product scope to reduce

the chances of failure and increase the chances of repeated successes. In other words, in an adaptive organization, the organization adapts to the user experience.

Organizations make technology for some reason, so user experience is implicitly included in all technology creation. However, explicitly basing every stage of technology development on user experience models is a relatively new concept. Though many organizations claim to be customer centered, in practice few product management practices actually make it the center of all of their activities. Most concentrate the examination of user and organizational needs at the beginning of a project (often called the "requirements gathering" phase) or at the end (the "evaluation" phase). Those are not the only options. Projects starting from scratch, where the technology is new and the development team is flexible, can use agile software development methods that introduce user-experience knowledge throughout the development process. However, mature products with long-established processes, attitudes, and methods often make starting from scratch impossible. This situation requires a different mix of techniques.

Agile User Experience Development

Henry Ford called his 1907 car the "Model T" because there was a model S before it, and a model R before that, all of the way back to the first Model A in 1903 (the 1928 Model A was a conscious rebranding to evoke a new philosophy of building cars). In other words, Henry Ford failed 20 times over the course of four years at making a successful passenger car. He iterated, based on feedback, on the idea until he found the correct combination of factors.

Iteration based on feedback is the core philosophy behind a family of software management practices called "agile software development." Agile development does not require detailed research and design specifications up front or paper trails and sign-offs throughout. Instead, agile methods focus on extensive communication, rapid iteration, and continuously collecting information and adjusting to it, rather than trying to plan the entire process.

As Highsmith (2002) described:

> Agility isn't a one-shot deal that can be checked off the organizational initiative list. Agility is a way of life, a constantly emerging and changing response to business turbulence. Critics may counter, "Agility is merely waiting for bad things to happen, then responding. It is a fancy name for lack of planning and ad hoc-ism." But Agile organizations still plan; they just understand the limits of planning. (p. 16)

Agile methodologies, which include Extreme Programming (Beck & Andres, 1999), Scrum (Schwaber & Beedle, 2001), and Crystal Clear (Cockburn, 2004), do not explicitly incorporate collecting and interpreting user-experience knowledge; rather, they continuously adapt to all new information.

Larman (2003) defined a set of core agile practices, summarized in Table 46.11.

TABLE 46.11. Agile Development Practices, Adapted from Larman (2003)

Iterative development
Risk-driven and client-driven
Timeboxing
Adaptive development
Evolutionary requirements

Iterative development. Larman (2003, p. 9) said, "An approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence. Each iteration is a self-contained mini-project composed of activities such as requirements analysis, design, programming, and test."

Iterations are typically from one to four weeks, with the goal of delivering "a stable, integrated, and tested partially complete system" (Larman, 2003, p. 10) with every iteration. In other words, an always-functioning system acquires functionality, in contrast to a collection of parts assembled at the end. The user experience frames the scope of each activity and sets priorities between them. The increments of functionality come from knowing the organization's goals for the product, from knowing the needs and values of the end-user audience, and from a negotiated balance between the two.

The most important elements to end users, which satisfy long-term goals of the organization, can be focused on first. For example, one project's first iteration focused on the interface for letting users retrieve information, even though there was no back end database. That interface was the core to meeting the user and business goals and had to be right.

Iteration does not need to start with functionality. Before programmers write any code or interface designers create screen designs, initial iterations can explore the audience's values and reactions with lightweight prototypes. For example, industrial design as practiced by design firm IDEO (Kelley & Littman, 2001) is highly iterative. Key interactions (as determined by research into end user and company goals) are prototyped, evaluated, and refined repeatedly before engineering technology begins. At the Rhode Island School of Design, "looks-like" and "works-like" prototypes differentiate the user experience from technological capabilities (Cottam, 2004). Returning to an earlier example, exploratory iterations on the children's art product website with end-user participation could have revealed that content meant for educators confused parents and children, or they could have revealed that educator content gave the website added authority.

Risk-driven and client-driven. Larman (2003, p. 12) said, "Risk-driven iterative development chooses the riskiest, most difficult elements for the early iterations [and] the choice of features for the next iteration comes from the client—whatever they perceive as the highest business value to them."

Treating both the end user experience of products and the user experience of organizations as parts of the same idea helps select among potential technological solutions. For example, a company making software for transportation logistics spent a calendar year, many developer years, and millions of dollars developing a complex feature that allows the system to send signal when cargo enters or leaves a certain geographic area. It sounds like a good idea, but it took much longer than initially

estimated and several years after its launch, customers had not broadly adopted it. User research with prototypes would have revealed that the technology did not fit work practices and that the business relationships did not support the information in the form provided. In other words, it does not solve a problem that people feel they have, and their business systems (including their business software) cannot use the information. Although organizational desire was high, for customers the risk of not doing it was low, thus the choice was neither risk-driven nor client-driven.

Timeboxing. Agile methodologies depend on being able to query a customer who interprets the user experience for the developers and makes priority trade-offs (for example, when something turns out to be harder than previously imagined). "Fixing the iteration end date and not allowing it to change" (Larman, 2003, p. 13) allows scheduled user research and organizational priority review. A regular research and release plan allows for much easier integration of the results of user research into development process. Such timeboxing allows the customer to plan for user research so that the results of the research become available when questions arise. For example, an Internet search engine with one-week iterations had a three-week research cycle. The research answered user experience questions posed by the developers, and upcoming features were prototyped and tested before expending any programming resources.

Adaptive development and evolutionary requirements. All of these practices boil down to two key concepts: development practices dynamically adapt to new information and requirements for the product change as knowledge about the user experience increases.

Introducing User Experience Into an Existing Process

We have to return to our entrepreneurial roots. (Highsmith, 2002)

There had been a growing sense among the Directors of the Product Management Group that there was a diminishing atmosphere of innovation within the group. (Fraser, 2005)

[To launch a brand new product] every mindset, timeline, and assumption had to be challenged. (Pellican & Homier, 2005)

Organizations regularly have crises where their leadership feels they have lost the ability to innovate. Rediscovering innovation is not impossible, merely difficult. Understanding the user experience is returning to an organization's entrepreneurial, innovative roots. Once, the organization had insight into the end-user needs and was able to balance those with its needs to be come successful. It lost the ability to be innovative when it lost that perspective.

No existing organization can change all its practices overnight. People need to be convinced at both the organizational and individual levels that there is value in change. On a new proj ect, it is easier to introduce new ways of creating, but everyone hates change when there is momentum. Forcing people to change in that situation almost never works, but introducing practices that make people's lives better and move the overall practice in the right direction sometimes works (sadly, there are no guarantees).

Development practices that expose the organization to user experience ideas lay the foundation for a gradual shift in perspective. The following practices are not in any order. They are ideas that seem to help organizations change from the inside out.

Get a senior manager champion. The preconditions to a successful change are the recognition of a need to change and the authority to make changes. Those with ultimate responsibility for the success of a product need to know that user-experience research is a key business process. Embracing the idea that the members of an organization are not representative of its audience has to start from the top, with the recognition by someone in a high-level position that they cannot manage by merely extrapolating from their experience. Make senior managers watch end user research is an effective and dramatic exercise that demonstrates the difference between the perception of a product inside and outside an organization. Watching someone struggle with a flagship application wins people over to the idea that maybe they do not know everything about how their end users view the world and the product. However, even without this, enlightened managers realize there is value (both for the organization as a whole and for their careers) in researching and codifying the user experience. Such managers make excellent champions within the organization. They are experts in the organization's needs and can serve as guides and translators, communicating issues to other managers, framing ways to justifying a practice that has no immediate return on investment, providing a voice of authority, and making resources available for the pursuit of such research. Few projects document their reliance on this relationship, but it is critical in nearly every successful organizational change. Hollinger (2005) said, "We requested management buy-in early on to be able to treat user experience defects the same as any other product defects. . . . This explicit support from a senior manager on the project was critical to our success in this area."

Work within existing processes. It is easy to dismiss new ideas as unworkable within the structure of an existing practice. Contextualize new practices in terms of the existing development process. For example, one UI design team first introduced a new practice—the keeping of user-experience scorecards—but when the practice's value became clear, the traditional process owners took over:

Although the design team produced the scorecards, the release management team eventually took over "enforcement", pushing teams to turn their yellow and red bubbles to green. This "mainstreaming" of the reviews resulting in significantly more user experience bugs being fixed. (Hollinger, 2005, p. 9)

This allowed the team to gradually introduce the practice while simultaneously showing its value. As the user experience spans technical and organizational practices, integrating it into both of those worlds produces the broadest effects. It can start by involving representatives from business units in the HCI research

process. Leadley, Pao, and Douglas (2005, p. 6) said, "You have to allot time and budget for the Business Analysts to attend user interviews and usability testing."

Later, it can progress to a more integrated approach:

> Getting the entire marketing, engineering, and quality assurance team to watch customer-interview tapes and call the customers themselves was a wonderful achievement. All teams now require this to be part of a new product process. (Pellican & Homier, 2005, p. 13)

Another way is to use familiar tools to represent user experience ideas. For example, Hollinger (2005) treated unmet user experience needs as "bugs" and used the internal bug-tracking system to keep track of them.

Make small, but well-publicized changes. Persistent internal marketing is crucial to wide-scale adoption of user experience ideas. Beginning with small projects, such as usability tests of existing projects, every report, presentation, and discussion can highlight insights gleaned from user experience research and analysis, linked to organizational goals. For example, one group chose to share their findings with the executive staff. Hollinger (2005) said, "The scorecards were presented to vice presidents in the development organization at release status meetings, adding legitimacy to user experience being an integral part of release quality decisions." (p. 9)

Internal marketing should be an integral part of a plan for changing a development culture. In a plan developed by Leadley et al. (2005; see Table 46.12, p. 4), most of the effort is devoted to marketing within the company, using a number of methods.

Such an extended effort takes patience, persistence, and resources, as acknowledged by Leadley et al. (2005, p. 5): "The 'sales' effort by way of 'dog and pony shows' absorbed as much of our time as defining standards, building elements and designing our website!"

However, internal marketing is inexpensive relative to failed products, and the potential benefits of these methods can be justified by comparing to one failed product launch, six months of delayed adoption, or another appropriate metric.

Make developers' lives easier with user experience. Technology development always happens under severe time constraints, and time-pressured developers resist additional work. It is one thing to communicate that user-experience research and analysis increases chances of long-term success, but demonstrating how it reduces work is even better.

One effective way to win over developers is to give them more freedom and reduce the amount of paperwork in their lives. Replacing documents with tools that embody good practices in code means that developers do not feel pressured to memorize complex standards or reinvent techniques. Apple Computer (n.d.) successfully enables developers to create consistent interfaces by backing up rules with a toolbox of interface elements and development tools that make it easier to follow the rules than to break them. Similarly, PBS created a set of "widgets" (Public Broadcasting System, 2004) that make it easier for member stations to conform to a uniform organizational standard. Leadley et al. (2005) and Kuniavsky and Raghavan (2005) created templates that included the code to generate interaction elements that conformed to end-user and organizational goals. "A UI Library with 9 templates and 40 elements, a high-level methodology, and a guide for how to use our system. . . . For both templates and elements we provided the HTML code and an HTML-rendered version of the item" (Leadley et al., 2005, p. 8).

TABLE 46.12. Timeline of Allianz User-Centered Practice Introduction (Leadley et al., 2005)

Year	Process	Number of Projects Involved
2001	Personas Dog & pony shows	4
2002	Usability testing UX Training 1st & 3rd Thursdays Dog & pony shows	10
2003	Usability testing JSP Benchmarking Case studies Prototyping UX Training JSP Training 1st & 3rd Thursdays	15
2004	Contextual inquiry Usability testing UX Training UX Thursdays SIG Presentations	17
2005	Contributions to Software Development Lifecycle Governance Standards Accountability of Enterprise usability Brand standards UX Thursdays	19

CONCLUSION

When an organization creates technology, it embodies in a product its idea for a solution to an end-user problem, with the goal that this will ultimately help the organization itself. HCI is how the end user interacts with the product, but this symbiotic relationship between the end users and the organization lies at the core of how that interaction is structured. Understanding the user experience, therefore, is the process of understanding the end-user needs and the organization needs with the goal of maximizing the benefit to both.

This is true regardless of whether the product is destined for a broad consumer market or an internal tool. Unfortunately, most methods still treat the interaction of humans with computers and the interaction of the product with the organization as different. In fact, developing the user experience is the whole of technology creation. Emotional, social, and organizational needs make up the fabric in which HCI exists. Without them, there would be no computers and no reason for humans to interact with them.

References

Agarwal, R., & E. Karahanna. (2000). Time flies when you are having fun: Cognitive absorption and beliefs about information technology usage. *MIS Quarterly, 24*(4), 665–694.

Andelman, D. A. (2005, October 5). A better way to search? Retrieved April 5, 2001, from Forbes.com

Apple Computer. (n.d.). Interface builder documentation. Retrieved December 1, 2004, from http://developer.apple.com/tools/interfacebuilder.html

Arnall, T. (2001). Mobile interaction design case study. Retrieved October 30, 2005, from http://www.elasticspace.com/2001/06/mobile-interaction-design-case-study

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Reading, MA: Addison-Wesley.

Bellotti, V. (1999). Personal communication.

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., et al. (1992, November). Ethnographically informed systems design for air traffic control. *Proceedings of the Conference on Computer Supported Cooperative Work*, Toronto, Canada, 123–129.

Berg, S., Taylor, A. S., & Harper, R. (2003, April). Mobile phones for the next generation: Device designs for teenagers. *Proceedings of the Conference on Human Factors and Computing Systems CHI 2003*, Fort Lauderdale, FL, pp. 443–440.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems.* San Francisco: Morgan Kaufmann

Buckman, R. (2002, June 14). Microsoft's cable-TV miscues turned into a costly lesson. *Wall Street Journal*.

Cockburn, A. (2004). *Crystal clear: A human-powered methodology for small teams.* Reading, MA: Addison-Wesley.

Cottam, M. (2004, April). Reuse for new uses. *Presentation at ICSID Forum,* CHI 2004, Vienna, Austria.

Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 14*(3), 319–340.

Elliot, K., Neustaedter, C., & Greenberg, S. (2005). Time, ownership and awareness: The value of contextual locations in the home. *Proceedings of the Ubicomp 2005 conference*, Tokyo, Japan, pp. 251–268.

European Information Technology Observatory. (2002). *Yearbook* (10th ed.). Frankfurt: Author.

Fraser, J. (2005, November). Inspired innovation—How Corel is drawing upon employees' ideas for user focused innovation. *Proceedings of DUX05: Conference on Designing for User eXperience*, San Francisco, CA, Electronic proceedings, ISBN 1–59593–250–X.

Garrett, J. J. (2000). *Elements of user experience.* Retrieved October 10, 2005, from http://www.jjg.net/elements/

Garrett, J. J. (2002). *Elements of user experience.* Indianapolis, IN: New Riders Press.

Haeckel, S. (1999). *Adaptive enterprise: Creating and leading sense-and-respond organizations.* Boston, MA: Harvard Business School Press

Hassenzahl, M. (2003). The Thing and I: Understanding the relationship between user and product. In M. A. Blythe, A. F. Monk, K. Overbeeke, & P. C. Wright (Eds.), *Funology: From usability to enjoyment*. Dordrecht, the Netherlands: Kluwer Academic.

Highsmith, J. (2002). *Agile software development ecosystems.* Reading, MA: Addison-Wesley.

Hollinger, M. (2005, November). A process for incorporating heuristic evaluation into a software release. *Proceedings of DUX05: Conference on Designing for User eXperience*, San Francisco, CA, Electronic proceedings, ISBN 1–59593–250–X.

Kelley, T., & Littman, J. (2001). *The art of innovation: Lessons in creativity from IDEO, America's leading design firm.* New York: Doubleday.

Krueger, R. A. (1988). *Focus groups: A practical guide for applied research.* Newbury Park, CA: Sage Publications.

Kuniavsky, M. (2003). *Observing the user experience: A practitioner's guide to user research.* San Francisco, CA: Morgan Kaufmann.

Kuniavsky, M., & Raghavan, S. (2005, November). Guidelines are a tool: Building a design knowledge management system for programmers. *Proceedings of DUX05: Conference on Designing for User eXperience*, San Francisco, CA, Electronic proceedings, ISBN 1–59593–250–X.

Larman, C. (2003). *Agile and iterative development: A manager's guide.* Reading, MA: Addison-Wesley.

Leadley, B., Pao, H., & Douglas, S. (2005, November). Creating a user experience culture at a nonsoftware company. *Proceedings of DUX05: Conference on Designing for User eXperience*, San Francisco, CA, Electronic proceedings, ISBN 1-59593-250-X.

Madison, D. S. (2005). *Critical ethnography: Method, ethics, and performance.* Thousand Oaks, CA: Sage Publications.

Maslow, A. H. (1943). A theory of human motivation. *Psychological Review, 50*, 370–396.

Millen, D. (2000, August). Rapid ethnography: Time deepening strategies for HCI field research. *Proceedings of DIS 2000: Designing Interactive Systems* , New York, NY, pp. 280–286.

Nass, C., & Reeves, B. (1996). *The media equation: How people treat computers, televisions, and new media as real people and places.* Cambridge, MA: Cambridge University Press.

Norman, D. A. (1998). *The invisible computer.* Cambridge, MA: MIT Press.

Ortony A., Norman, D. A., & Revelle, W. (2005). Affect and proto-affect in effective functioning. In J. M. Fellous & M. A. Arbib (Eds.), *Who needs emotions? The brain meets the machine* (pp. 173–202). New York: Oxford University Press.

Pellican, S., & Homier, M. (2005, November). Customer driven innovation: Quicken(r) rental property manager. *Proceedings of DUX05: Conference on Designing for User eXperience*, San Francisco, CA, Electronic proceedings, ISBN 1–59593–250–X.

Pine, B. J., & Gilmore, J. H. (1999). *The experience economy.* Boston, MA: Harvard Business School Press.

Postrel, V. (2003). *The substance of style: How the rise of aesthetic value is remaking commerce, culture and consciousness.* New York: HarperCollins.

Public Broadcasting System. (2004). *Best practices for PBS member stations.* Retrieved December 1, 2005, from http://www.pbs.org/remotecontrol/bestpractices/

Rafaeli, Y., & Vilnai-Yavetz, I. (2004). Emotion as a connection of physical artifacts and organizations. *Organizational Science, 15*, 6.

Saffer, D. (2002, June 3). Building brand into structure. *Boxes and Arrows.* Retrieved October 28, 2005, from http://www.boxesandarrows.com/archives/building_brand_into_structure.php

Sawhney, M. (2003, July 1). Fundamentals of Value. *CIO Magazine.*

Schmitt, B. H. (2003). *Customer experience management: A revolutionary approach to connecting with your customers.* San Francisco, CA: John Wiley & Sons.

Schwaber, K., & Beedle, M. (2001). *Agile software development with SCRUM.* Upper Saddle River, NJ: Prentice Hall.

Spool, J. (2005, August 10). Re:[Sigia-L] Length of nav Labels. Message posted to Sigia-L electronic mailing list, archived at http://www.info-arch.org/lists/sigia-l/0508/0157.html

TiVo. (2005, August 24). Press release.

U.S. General Accounting Office. (1996). *Content analysis: A methodology for structuring and analyzing written material* (GAO/PEMD-10.3.1). Washington, DC: Author.

Vredenburg, R., Isensee, S., & Righi, C. (2001). *User-centered design: An integrated approach*. Upper Saddle River, NJ: Prentice Hall.

Wikipedia. (2005a). *Ford Model T.* Retrieved October 10, 2005, from http://en.wikipedia.org/wiki/Ford_Model-T

Wikipedia. (2005b). *Ford Model A.* Retrieved October 10, 2005, from http://en.wikipedia.org/wiki/Ford_Model_A

Wikipedia. (2005c). *Customer relationship management.* Retrieved October 29, 2005, from http://en.wikipedia.org/wiki/Customer_relationship_management

Wikipedia. (2005d). *Customer experience management.* Retrieved October 29, 2005, from http://en.wikipedia.org/wiki/Customer_experience_management

Wikipedia. (2005e). *Out of box experience.* Retrieved October 30, 2005, from http://en.wikipedia.org/wiki/Out-of-box_experience.

Wixon, D. R., Ramey, J., Holtzblatt, K., Beyer, H., Hackos, J., Rosenbaum, S., et al. (2002, April). Usability in practice: Field methods, evolution and revolution. *Proceedings CHI 2002: Proceedings of the Conference on Human Factors and Computing Systems* , Minneapolis, MN, pp. 880–884.

Zhang, P., & Li, N. (2005). The importance of affective quality. *Communications of the ACM, 9*, 105–108.

◆47◆

REQUIREMENTS SPECIFICATIONS WITHIN THE USABILITY ENGINEERING LIFECYCLE

Deborah J. Mayhew, PhD
Deborah J. Mayhew & Associates

Introducing Requirements Specifications **918**
Phase 1: Requirements Analysis 918
User profile 918
Contextual task analysis 918
Usability goal setting 918
Platform capabilities/constraints 918
General design guidelines 919
Phase 2: Design/Testing/Development 919
Level 1 Design 919
Work reengineering 919
Conceptual model design 920
Conceptual model mockups 920
Iterative conceptual model evaluation 920
Level 2 Design 920
Screen design standards 920
Screen design standards prototyping 920
Iterative screen design standards evaluation 920
Style guide development 920
Level 3 Design 920
Detailed user interface design 920
Iterative detailed user interface design evaluation 920
Phase 3: Installation 920
User feedback 920
Focusing on Requirements Analysis **920**
The user profile 920
Contextual task analysis 921
Motivating and Justifying Requirements Specification **923**
User Requirements 923
Work Environment Requirements 923
Task Requirements 924
References **926**

INTRODUCING REQUIREMENTS SPECIFICATIONS

Three key ingredients are necessary to ensure that usability is achieved during product development:

- Application of established design principles and guidelines;
- A structured methodology for design;
- Managerial and organizational techniques.

At this point in the history of the field of human-computer interaction, there are well-established software design principles and guidelines available, based on objective research and reported in the literature. Many of these principles and guidelines are enumerated throughout different chapters in this book. Development organizations need to have staffs which are fluent in these design guidelines participate in design efforts, so this general accumulated knowledge will find its way into their products.

However, just having a design guru on board does not guarantee that design principles and guidelines will find their way into products. Design is complex, and there simply is no cookbook approach to design that can rely on general principles and guidelines alone. Development organizations also need structured methods for achieving usability in their products.

Similarly, a well-structured and documented design methodology must be introduced and managed—it does not happen by itself. Thus, managerial and organizational techniques must be applied to ensure that the design methodology is followed and includes the application of well-established design principles.

Even when good management practices are being applied, either of the remaining two ingredients alone—design guidelines or design methods—is necessary, but not sufficient. Optimal design cannot be accomplished by the systematic application of generic guidelines alone, because every product and its intended set of users are unique. Design guidelines must be tailored for and validated against product-unique requirements, and this is what the structured methods accomplish.

Conversely, applying structured methods without also drawing on well-established design principles and guidelines is inefficient at best and may simply fail at worst. Without the benefit of the initial guidance of sound design principles during first passes at design, a particular project with its limited resources may simply never stumble upon a design approach that works. For example, formal usability testing is a valuable and objective method for uncovering usability problems. However, without a clear understanding of basic design principles and guidelines, as well as unique requirements data, solving those problems after they have been identified will not be easy or likely.

Previous sections and chapters in this book refer to a broad variety of design principles and guidelines. This section (section A in Part IV) addresses the need for methodology and provides chapters that address a variety of techniques that can be applied during the development process to achieve usability in product design. This chapter sets the stage for this section and, in particular, for the subsection on one stage of the development process: requirements specification.

The Usability Engineering Lifecycle (Mayhew, 1999) documented a structured and systematic approach to addressing usability within the product-development process. It consists of a set of usability engineering tasks applied in a particular order at specified points in an overall product-development lifecycle.

Several types of tasks are included in the Usability Engineering Lifecycle:

- Structured usability-requirements analysis tasks;
- An explicit usability goal-setting task, driven directly from requirements-analysis data;
- Tasks supporting a structured, top-down approach to user interface design that is driven directly from usability goals and other requirements data;
- Objective usability evaluation tasks for iterating design toward usability goals.

Figure 47.1 represents in summary, visual form, the Usability Engineering Lifecycle. The overall lifecycle is cast in three phases: Requirements Analysis, Design/Testing/Development, and Installation. Specific usability-engineering tasks within each phase are presented in boxes, and arrows show the basic order in which tasks should be carried out. Much of the sequencing of tasks is iterative, and the specific places where iterations would most typically occur are illustrated by arrows returning to earlier points in the lifecycle. Brief descriptions of each lifecycle task follow.

Phase 1: Requirements Analysis

User profile. A description of the specific user characteristics relevant to user interface design (e.g., computer literacy, expected frequency of use, level of job experience) is obtained for the intended user population. This will drive tailored user interface design decisions and identify major user categories for study in the Contextual Task Analysis task. This volume includes a set of chapters (Part V) dedicated to exploring the unique needs and requirements of different types of user populations.

Contextual task analysis. A study of users' current tasks, workflow patterns, and conceptual frameworks is made, resulting in a description of current tasks and workflow, and understanding and specification of underlying user goals. These will be used to set usability goals and drive Work Reengineering and user interface design. This section of this volume includes several chapters on topics related to task analysis (Chapters 48, 49, 50).

Usability goal setting. Specific qualitative goals reflecting usability requirements are developed, extracted from the User Profile and Contextual Task Analysis. In addition, quantitative goals (based on a subset of high-priority qualitative goals) are developed, defining minimal acceptable user performance and satisfaction criteria. These usability goals focus later design efforts and form the basis for later iterative usability evaluation.

Platform capabilities/constraints. The user interface capabilities and constraints (e.g., windowing, direct manipulation, screen size, color, etc.) inherent in the technology plat-

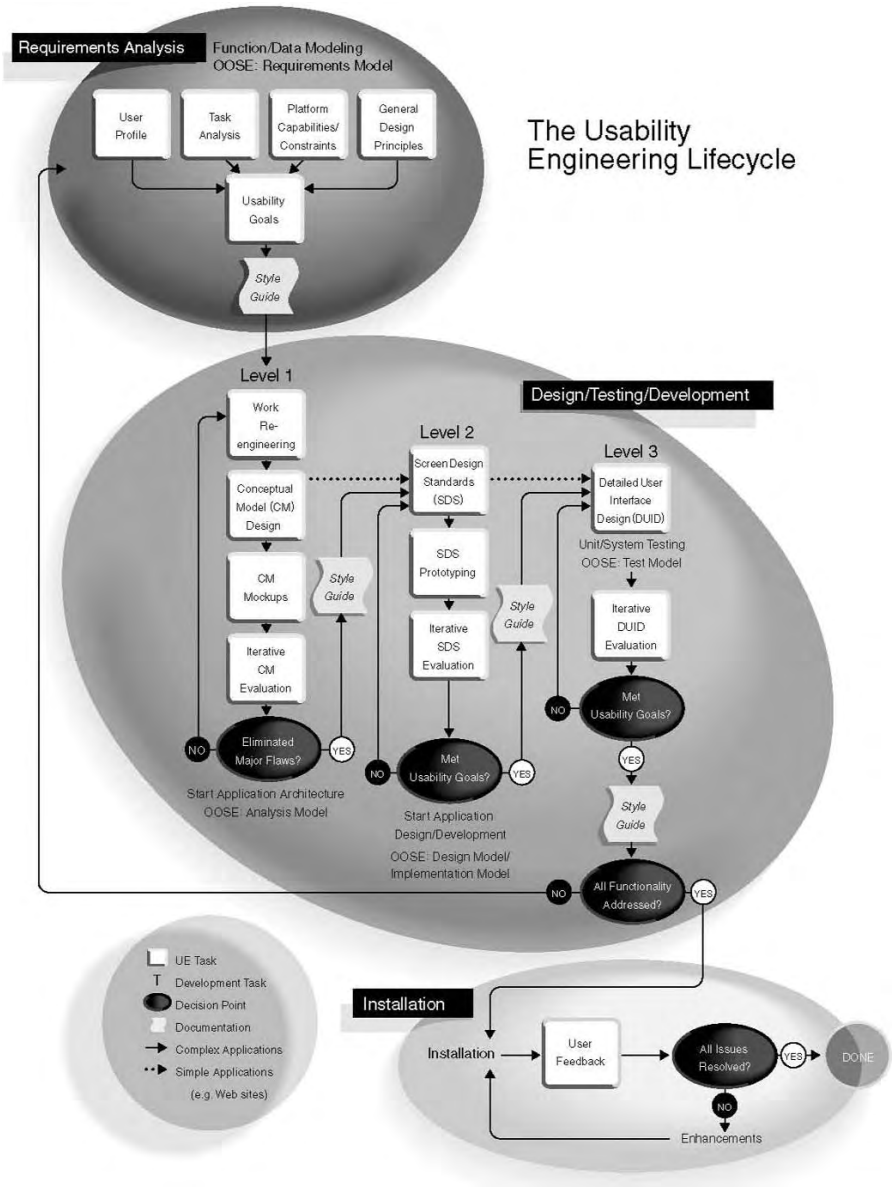


FIGURE 47.1. The Usability Engineering Life Cycle. (Illustration taken from Mayhew, 1999).

form chosen for the product (e.g., Apple Macintosh, MS Windows, product-unique platforms) are determined and documented. These will define the scope of possibilities for user interface design. This volume includes chapters which describe platform capabilities that support many relatively new and innovative types of user interfaces, including multimedia (chapter 20), adaptive interfaces and agents (chapter 22), mobile devices and ubiquitous computing (chapter 23), tangible interfaces (chapter 24), and wearable computers (chapter 15).

General design guidelines. Relevant general user interface design guidelines available in the usability engineering literature are gathered and reviewed. They will be applied during the design process to come, along with all other project-specific information gathered in the previous tasks.

Phase 2: Design/Testing/Development

Level 1 Design

Work reengineering. Based on all requirements-analysis data and the usability goals extracted from it, user tasks are redesigned at the level of organization and workflow to streamline work and exploit the capabilities of automation. No user interface design is involved in this task, just abstract organization of

functionality and workflow design. This task is sometimes referred to as "Information Architecture."

Conceptual model design. Based on all the previous tasks, initial high-level design alternatives are generated. At this level, navigational pathways and major displays are identified, and rules for the consistent presentation of work products, processes, and actions are established. Screen design detail is *not* addressed at this design level.

Conceptual model mockups. Paper-and-pencil or prototype mockups of high-level design ideas generated in the previous task are prepared, representing ideas about high-level functional organization and Conceptual Model Design (see Beaudouin-Lafon and MacKay in this *Handbook* for a discussion of prototyping tools and techniques). Detailed screen design and complete functional design are not in focus here.

Iterative conceptual model evaluation. The mockups are evaluated and modified through iterative evaluation techniques such as formal usability testing, in which real, representative end users attempt to perform real, representative tasks with minimal training and intervention, imagining that the mockups are a real product user interface (see section C, Part VI in this volume for chapters on testing and evaluation). This and the previous two tasks are conducted in iterative cycles until all major usability "bugs" are identified and engineered out of Level 1 (e.g., Conceptual Model) design. Once a Conceptual Model is relatively stable, system architecture design can commence.

Level 2 Design

Screen design standards. A set of product-specific standards and conventions for all aspects of detailed screen design is developed, based on any industry and/or corporate standards that have been mandated (e.g., Microsoft Windows, Apple Macintosh, etc.), the data generated in the Requirements Analysis phase, and the product-unique Conceptual Model Design arrived at during Level 1 Design.

Screen design standards prototyping. The Screen Design Standards (as well as the Conceptual Model Design) are applied to design the detailed user interface to selected subsets of product functionality. This design is implemented as a running prototype.

Iterative screen design standards evaluation. An evaluation technique, such as formal usability testing, is carried out on the Screen Design Standards prototype, and then redesign/reevaluation iterations are performed to refine and validate a robust set of Screen Design Standards. Iterations are continued until all major usability bugs are eliminated and usability goals seem within reach.

Style guide development. At the end of the design/evaluate iterations in Design Levels 1 and 2, you have a validated and stabilized Conceptual Model Design, and a validated and stabilized set of standards and conventions for all aspects of detailed Screen Design. These are captured in the document called the product Style Guide, which already documents the results of requirements-analysis tasks. During Detailed User Interface Design, following the Conceptual Model Design and Screen Design Standards in the product Style Guide will ensure quality, coherence, and consistency—the foundations of usability.

Level 3 Design

Detailed user interface design. Detailed design of the complete product user interface is carried out based on the refined and validated Conceptual Model and Screen Design Standards documented in the product Style Guide. This design then drives product development.

Iterative detailed user interface design evaluation. A technique such as formal usability testing is continued during product development to expand evaluation to previously unassessed subsets of functionality and categories of users, and also to continue to refine the user interface and validate it against usability goals.

Phase 3: Installation

User feedback. After the product has been installed and in production for some time, feedback is gathered to feed into design enhancements, design of new releases, and/or design of new but related products.

FOCUSING ON REQUIREMENTS ANALYSIS

This section of the *Handbook* provides in-depth coverage of the different phases (and tasks within those phases) of the Usability Engineering Lifecycle. In particular, this subsection, Requirements Specification, provides chapters describing and discussing different tasks and techniques for requirements specification in some depth and from different perspectives. The goal of this chapter is to reinforce the importance of this phase of the lifecycle and to provide real-world examples of the benefits of conducting the kinds of techniques discussed in the later chapters of this subsection.

Three main things must be studied and understood to tailor design to support unique requirements:

- The users
- The users' tasks
- The users' work environment

In The Usability Engineering Lifecycle, the first is addressed in the task called the User Profile, and the second two are addressed in the task called Contextual Task Analysis.

The user profile. There is no single best user interface style or approach for all types of users. Specific interface design alternatives that optimize the performance of some types of users may actually degrade the performance of other types of users. For example, an infrequent, casual user needs an easy-to-learn and easy-to-remember interface, but a high-frequency expert user

needs an efficient, powerful, and flexible interface. These are not necessarily the same thing. Similarly, a highly skilled typist might perform better with a keyboard-oriented interface, whereas a low-skill typist might do better with a graphical user interface in which point-and-select interaction replaces keyboarding.

Many user populations have very specific special needs that impact their ability to interact with software applications. Chapters in this section cover the special characteristics of the elderly (see chapter 39), children (see chapter 40), cross-cultural audiences (see chapter 18), and users with various kinds of physical and perceptual disabilities (see chapters 42–43).

Unless designers know the specific characteristics of a population of users (e.g., expected frequency of use, level of typing skill, special needs and constraints, etc.), they cannot make optimal user interface design decisions for them. The purpose of a User Profile is thus to establish the general requirements of a category of users in terms of overall interface style and approach.

One relatively new technique for gathering, documenting, and applying user profile data in user interface design is the use of "personas" (Pruitt & Adlin, 2006; see also Adlin & Pruitt in this volume). Personas are realistic and detailed descriptions of imaginary people who represent all the characteristics, skill sets, goals, top tasks, responsibilities, tools, pain points, and so on of a category of users. They are used to drive discussion and design all throughout the product-development lifecycle, from product conception to user acceptance testing. They help keep the focus on users' needs and goals, and give stakeholders a common language with which to consider, discuss, and evaluate product-design ideas. Another older, but still effective, way to keep the user perspective in product design is participatory design, which builds users right into the design team (see chapter 54 in this volume).

Besides understanding individual users of different categories, understanding the characteristics of groups and communities is another aspect of user profiling that becomes important in the design of groupware and products aimed at online communities and computer-supported cooperative work. This aspect of requirements analysis is addressed by Olson and Olson (chapter 27) in this volume.

The User Profile task fits into the overall Usability Engineering Lifecycle as described as follows:

- The User Profile task is the first task in the Usability Engineering Lifecycle;
- The User Profile task will feed directly into the Contextual Task Analysis task by identifying categories of users whose tasks and work environment must be studied in that later task;
- The User Profile task will feed directly into the Usability Goal Setting task in that usability goals are in part driven directly by user characteristics (e.g., a low frequency of use indicates a need for ease-of-learning and remembering). Thus, different usability goals will be extracted from the profiles of different categories of users;
- Ultimately, the User Profile task will have a direct impact on all design tasks, which are focused on realizing usability goals, in turn based in part on User Profiles;
- The User Profile task will also drive the selection of usability-evaluation issues and test users;
- Output from the User Profile task will be documented in the product Style Guide.

The User Profile task fits into the underlying software development methodology as follows:

- The User Profile task can occur in parallel with, overlapping with, or following the development of the Requirements Model in the Analysis Phase in Object-Oriented Software Engineering (OOSE), or function and data modeling in the requirements phase of a traditional rapid prototyping methodology. It could either define Actors for the Requirements Model or take the definition of user categories from the Requirements Model as its starting point;
- The User Profile task (along with all other Usability Engineering Lifecycle Requirements Analysis tasks) should occur prior to the development of the Analysis Model in the Analysis phase of OOSE (or the application architecture design in a traditional rapid prototyping methodology).

Contextual task analysis. The purpose of the Contextual Task Analysis task is to obtain a user-centered model of work as it is currently performed (e.g., to understand how users currently think about, talk about, and do their work in their actual work environment). Ultimately, the reason for this is that when designing a new product and its user interface, it is important to find an optimal compromise or tradeoff between three goals:

- Realizing the power and efficiency that automation makes possible;
- Reengineering work processes to more effectively support identified business goals;
- Minimizing retraining by having the new product tap as much as possible into users' existing task knowledge, and maximizing efficiency and effectiveness by accommodating human cognitive constraints and capabilities within the context of their actual tasks.

Traditionally, in software-development methodologies, the most focus is put on the first goal, and some of the second goal and most of the third are lost, because designers and developers never really understand the users' work and their current work models. A great deal of work reengineering occurs in application design, and only some of it serves the first two goals. Much of it is unnecessary and not useful, and it results in unnecessary training and usage burdens on the user. The third goal—minimizing the training overhead and maximizing efficiency and effectiveness—cannot be factored in unless designers have a clear picture of users' current work models: how they do their work in the realities of their everyday work environment, and how they think about it and talk about it.

Traditional systems analysis usually (but not always) results in the inclusion of all required data and low-level functions, and structures them in a robust implementation architecture. Without a truly user-centered approach, however, it often fails to organize and present that data and functionality in a manner that supports and optimizes the work performance of real users in their real work environment. This missing piece is the whole point of a Contextual Task Analysis.

Thus, the purpose of this task is to supplement more traditional types of systems analyses to define usability requirements and to point toward ways of meeting those requirements. Then,

in later tasks, these requirements can be applied directly to making user interface design decisions.

Contextual Task Analysis consists of the following basic steps:

- Gathering background information about the work being automated;
- Collecting and analyzing data from observations of and interviews with users as they do real work in their actual work environment;
- Constructing and validating models of how users currently think about and do their work.

A central, key step in the Contextual Task Analysis task is the second step, sometimes referred to as "contextual observations/interviews." Here, the idea is that analysts must observe and interview users in their real-life work context to understand their work and discover their work models. Only then can designers structure and present functionality in an application user interface in a way that taps into users' current work models and optimally supports their tasks.

An abstract modeling of users' tasks, which is the focus of more traditional types of business and systems analysis, does not typically take into consideration key aspects of actual workflow and key aspects of the users and their work environment, and simply does not support this goal. Another way of putting this is that traditional systems analysis models work in the abstract, without considering the basic capabilities and constraints of human information processing, the particular characteristics of the intended user population, the unique characteristics of the work environment, and how users themselves model, carry out, and talk about their tasks. In addition, the usual approach in a traditional business/systems analysis is to decompose high-level units of work into low-level functions. In the process, much of how the work is actually carried out is lost. The result of this analysis approach is often systems that provide all necessary functionality, but in an organization and presentation that simply does not support the natural flow of work and address the current "points of pain" for individual users.

Based on an analysis of direct observations, models can be constructed that represent not the work in the abstract from a systems point of view, but instead represent how users currently think about, talk about, and actually carry out their work (e.g., models reflecting the users' point of view). These models do not get directly designed into the application or its interface. They feed into only one of the goals referred to earlier, that of tapping into existing user knowledge, habit, and capabilities, and they are juggled with the other two goals of supporting general business goals and exploiting the power of automation. This juggling happens in a later task in The Usability Engineering Lifecycle called "Work Reengineering," but also often referred to as "Information Architecture."

One generic aspect of tasks that has become increasingly important with the explosion of e-commerce is the issues of security and privacy. Understanding users' requirements for and expectations of privacy, and making users feel confident in the security and privacy of personal information they are asked to divulge in the course of various kinds of online transactions, has become a significant part of the necessary research into user tasks, and the design of interactions to support them. This important new aspect of task analysis is covered by Karat, Karat, and Brodie (chapter 32) in this volume.

Contextual Task Analysis task fits into the overall Usability Engineering Lifecycle as described as follows:

- The User Profile task will feed directly into the Contextual Task Analysis task by identifying categories of users (e.g., actors) whose tasks must be studied;
- The Contextual Task Analysis task will feed directly into the Usability Goal Setting task by helping to identify different primary goals for different task types (use cases), and by identifying bottlenecks and weaknesses in current work processes that can be reduced through good user interface design;
- The Contextual Task Analysis task will feed directly into the Work Reengineering task. Current user work models are reengineered only as much as necessary to exploit the power of automation and contribute to explicit business goals. Current user knowledge and experience are exploited as much as possible to facilitate ease of learning and use;
- The Contextual Task Analysis task will be documented in the product Style Guide;
- Ultimately, the Contextual Task Analysis task will have a direct impact on all design tasks, and on the selection of usability testing and evaluation issues, as well as on the design of usability testing materials.

Contextual Task Analysis fits into the underlying software development methodology as follows:

- The Contextual Task Analysis task can occur in parallel with, overlapping with, or following the development of the Requirements Model in the Analysis Phase in OOSE (or function and data modeling in the requirements phase of a traditional rapid prototyping methodology). It could either identify Use Cases for the Requirements Model, or take the definition of Use Cases from the Requirements Model as its starting point for constructing Task Scenarios;
- The Contextual Task Analysis task (along with all other Usability Engineering Lifecycle Requirements Analysis tasks) should occur prior to the development of the Analysis Model in the Analysis phase of OOSE (or the application architecture design in a traditional rapid-prototyping methodology).

Currently, in the field of Usability Engineering, there is no well-established, universally applied, general, practical, and highly structured technique for performing a Contextual Task Analysis for the purpose of driving the user interface design for a product that has already been identified and scoped. It is more of an art than a science at this point, with each usability practitioner using his or her own informal approach. Recently, however, some structured techniques have begun to emerge, and experience with them has been reported in the literature (e.g., Beyer & Holtzblatt, 1998; Hackos & Redish, 1998). Later chapters in this book represent the latest forays into the thinking of these and other experts on this important task in the Usability Engineering Lifecycle.

MOTIVATING AND JUSTIFYING REQUIREMENTS SPECIFICATION

While the previous section set the stage for Requirements Specifications within the broader perspective of the whole Usability Engineering Lifecycle and underlying development process, this section provides motivation and justification for investing in Requirements Specifications tasks and activities, in particular through the reporting of "war stories" from the experience of myself and other practitioners. The war stories are divided into those relating to user, environment, and task requirements.

User Requirements

Following are two examples from my own experience of the importance of knowing your users.

For two of my clients, I first interviewed project team members (developers) to get a general sense of the user population. My purpose was to solicit input to the design of a User Profile questionnaire that I would later employ to solicit profile information directly from the users themselves.

In one case, the project team was convinced that their users would have a generally low level of familiarity with the Microsoft Windows platform they planned for their product. They were thus prepared to depart significantly from the Windows platform user interface standards in their product user interface. The User Profile questionnaire, however, revealed a generally high level of Windows experience. This, and the fact that the Windows user interface standard was a good fit for the application functionality, led me to strongly advise them to adopt the Windows standards as closely as possible. They were still interested in creating their own unique user interface, but early testing of several alternative designs that varied in how faithfully they followed Windows standards clearly showed that users learned much more quickly the more consistent the design was with Windows standards.

On the other project, the development team felt quite confident that users would generally have high levels of computer literacy. An extensive User Profile questionnaire of more than 800 users, however, revealed that only a very small percentage of potential users had any experience with computer software at all, let alone the Windows user interface standards. In this case, based on this User Profile data, we designed (and validated through testing) a highly simplified user interface that departed significantly from the Windows standards.

In both of these cases, two things are clear:

- Project team members often have serious misconceptions about the key characteristics of their users;
- These misconceptions could lead teams to design inappropriate user interfaces for those users.

Work Environment Requirements

It is also important to understand the environment in which users will be utilizing a product to carry out their tasks, because this environment will place constraints on how they work and how well they work.

By analogy, suppose a screwdriver is being designed and all that is known is the size of the screw head it must fit. So, something like a traditional screwdriver is designed, with the correctly sized blade. But, suppose it then turns out that the user needs to apply the screw from the inside of a narrow pipe to assemble some piece of equipment. Clearly, a traditional screwdriver will be useless in this work context.

Similarly, suppose a software application is being designed for a set of users, but the designers have never gone to the users' actual work environment. They assume a traditional office-like environment and design software that will work on a traditional workstation. But, suppose it then turns out that in the actual work environment, users are constantly in motion, moving all around the environment to get different parts of an overall job done. If software for a traditional workstation is designed, this will simply not work in this environment. Software that will run on a smaller and more portable device that can be carried around with the user, such as the units carried by UPS delivery staff, would be required instead.

In another example, suppose designers have never visited the user's workplace, and they assume users all work in closed offices. So, a system with voice input and output is designed. But, it then turns out that users work in one big open area with desks located right next to one another. The noise from all those talking people and workstations will create an impossible work environment, and most voice-recognition systems simply do not work with acceptable accuracy in a noisy environment. The point is, there are many aspects of the actual work environment that will determine how well a tool will work in that environment, and so the environment itself must be studied and the tool tailored to it.

A real example of the importance of understanding the users' work environment comes from a project I worked on with a large metropolitan police department. Requirements-analysis activities revealed that in the typical police station, the appearance of the interior is dark, run-down, and cluttered; the lighting is harsh and artificial; and the air is close and sometimes very hot. The noise level can be high, the work areas are cramped and cluttered, and the overall atmosphere is tense and high-pressured at best, chaotic and sometimes riotous at worst. These conditions most likely have a general impact on morale, and certainly will have an impact on cognitive functioning that, in turn, will impact productivity and effectiveness. A user interface must be carefully designed to support the natural and possibly extreme degradations of human performance under these conditions.

In addition, it was observed that in the noisy, stressful, and distracting work environment in a typical police station, users will frequently be interrupted while performing tasks, sometimes by other competing tasks, sometimes by unexpected events, unpredictable prisoners, and so forth. A user interface in such an environment must constantly maintain enough context information on the screen so that when users' attention is temporarily but frequently drawn away from their tasks, they can quickly get reoriented and continue their task without errors, and not have to backup or repeat any work.

Task Requirements

Besides the users themselves and the environments they work in, the tasks they do also have their own inherent requirements.

One compelling example of the need for a thorough understanding of users' tasks to achieve usable design comes from one of the earliest books on computer-human interaction (Rubinstein & Hersh, 1984, p. 26).

For example, recently a water district in Maine installed an online billing system so that when a customer called in, the clerk could quickly retrieve a record of usage and billing for that customer. The managers noticed that the clerks were less than happy with the new system. A little investigation revealed that with the manual system, the clerks would write pertinent information in the margins of the records, such as that one person pays on the 15th of the month rather than on the first, or that the meter reader should contact the neighbor across the street for access to a house. This informal information was critical to the smooth operation of the office, but because there was no provision for it in an official field of the payment-record form, the information was lost in the conversion to the online system. Understanding that there is informal as well as official information and recognizing the importance of the former would have reduced the disruption in work style.

Another good example of what can be missed by not understanding users' tasks is found in Coble, Karat, and Kahn (1997). This report described the use of task-analysis techniques to study the functional requirements of physicians for a clinical workstation:

> Before the . . . session started, a physician explained the purpose and details of each section in his office chart. . . . Later, when he was doing actual work in context, the person performing the . . . session noticed that the note he was looking at was written with red ink. She probed and the physician said it told him the previous encounter with this patient was a hospital visit. That fact told him he needed to review the hospital discharge summary and hospital laboratory results before entering the patient's exam room. *The physician was surprised that he had not mentioned that need before.* It was so ingrained in how he worked that he did not even process that highly relevant detail consciously anymore (p. 231). [Italics mine.]

In another example from my own experience, I was once performing a requirements analysis in a police department and observing police officers using a system of standardized paper forms to document property. I observed infrequent users of the forms struggling with them; there were many complex forms and many undocumented rules about how to fill them out. Later I observed very frequent users using them. The frequent users tended to ignore the forms initially and take freeform notes describing the property they had to document in a format very different from that required on the forms. They then transcribed their own notes onto the forms as required. It was clear from this observation (and from follow-up interviews with the frequent users) that the forms were not designed in a way that supported the users' task, and I learned a lot from how frequent users worked around the forms with their own format that helped in designing better online forms. A traditional systems analysis would typically not involve studying users actually using paper forms in their actual work, but would have instead taken the paper forms themselves as a description of the work to be automated. Problems with the forms would have been missed entirely and perpetuated in the online version of the task.

There are always a great many things like this that users simply will not think to report during an offsite interview or "focus group" that only emerge during in-context observations of people doing real work in their workplaces. Such things can have major implications for product user interface design.

Here is another example, this time from the context of a customer service application in an insurance company. In the navigational model illustrated in Fig. 47.2, the free text in normal intensity represents the basic user tasks identified from initial in-context observations with users. The bold text in boxes represents the hierarchy of groups that the design team developed, with labels they created—again, all based on initial task analysis activities. (This hierarchy represents only a partial set of the functionality required by a Customer Support Rep. For simplicity, not all groups or low-level tasks are included in the example.)

In contrast to the designer-generated model in Fig. 47.2, Fig. 47.3 shows a user-generated model. Again, the free text in normal intensity represents the low-level tasks presented to the user, each one on a separate index card. This time the bold text in boxes represents the groups that one user formed from the low-level tasks, with labels they suggested.

The differences between the designer-generated model and the user-generated model of these tasks are significant.

For example, the design team organized all types of Changes (e.g., Address, Beneficiary) under a single category ("Change Requests") within the category Sales Support, whereas this user distinguished between those changes that have to do with cus-

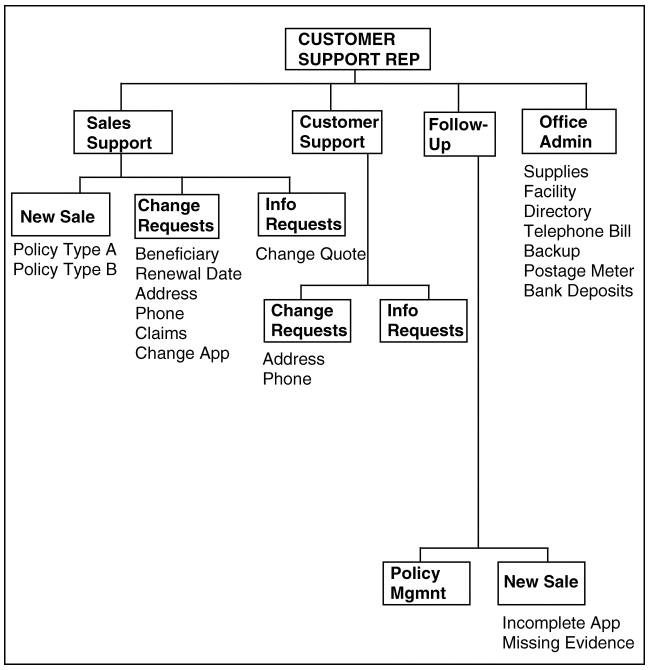


FIGURE 47.2. A Designer's Navigational Model. Admin = Administration; App = application; Mgmnt = management.

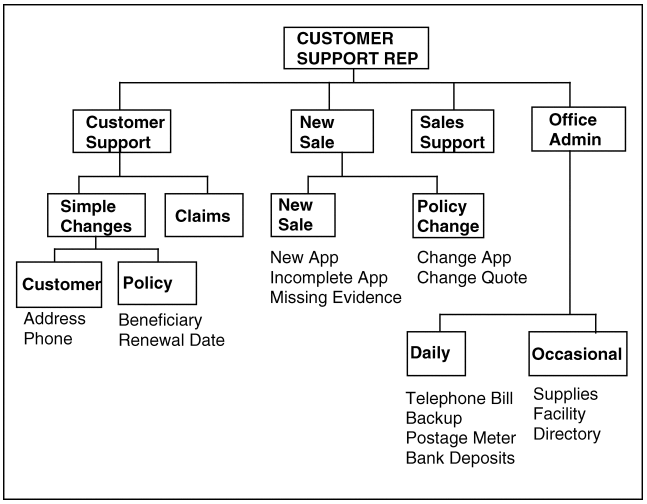


FIGURE 47.3. A User's Navigational Model. Admin = Administration; App = application.

tomer information and those that have to do with policy details, but located both under the category Customer Support rather than Sales Support. As it turned out, Customer Support staff can make simple changes to customer information on their own, but more complex paperwork and approvals are required for policy changes. Thus, these are two distinct categories of tasks to users. Even though requests for these types of changes often come from customers via sales agents, this user still regarded them as really reflecting Customer Support.

Similarly, the design team lumped all Office Administration tasks in one group, whereas this user distinguished between daily and occasional office tasks. Different people get assigned to these types of tasks, so, again, this is a meaningful distinction to users.

The designers imagined a category unto itself called Follow-Up, in which incomplete tasks of all types (e.g., Incomplete App) are all located, whereas this user considered follow-up activities as belonging with the original task types with which they are associated.

Finally, note that the user regarded New Sales (data entry of new policy applications) as its own category, whereas the designers had seen this as a subcategory of Sales Support—a very different perception.

One important thing to note in comparing this user's model of tasks with the designer's first pass at a model of tasks is that, for any set of low-level tasks, there may be a large number of very different "logical" task organizations possible, but there will only be a small number of rather similar ones that will make sense to users, given their actual work. A traditional systems analysis will decompose high-level functional requirements to all the corresponding low-level tasks, but it typically does not ensure that the task organization most consistent with the users' models of their tasks will be presented to the user in the user interface to a product intended to support those tasks. That is one of the main goals of a task analysis.

Also note that this user's model is most likely not exactly the same as other users' models, although in most cases, there will be a lot of similarity across individual users' models. It is necessary to consolidate all the sampled users' task models and generate one consolidated model (in the same format as the individual users' models, as illustrated in Fig. 47.3) that captures all the across-user commonalties as well as possible.

One last example from my recent experience illustrates the importance of studying users' work in depth to design the user interface to a software application that supports that work.

In this example, the application was a database system intended to support users in a government agency. The mission of the agency was to uncover and prosecute criminal behavior of a specific type. The job of the users of the intended system was to manage publicity for their agency in the media. These users needed to interact with the media to enhance reporting on criminal cases to use the media to help discourage criminal behavior and encourage public cooperation in reporting and investigating criminal behavior.

The database of information being provided to these users included information on criminal cases in progress relevant to this agency, prior press releases and news articles on relevant criminal cases, a "phonebook" of contact information for reporters and other contacts in the media, records of past communications with the media, and so forth. A card-sorting exercise revealed that users considered these categories of data types—Cases, Documents, Media Contact Information, Media Communications—to be distinct and meaningful categories that would provide a good foundation for the navigational structure of the application. However, what was not revealed by the card sorting exercise, but did become clear through other methods of studying users' work in context, was that when users searched for and found a particular item in one of these categories, such as a case, they then typically wanted to automatically see items in other categories related to the item they had initially looked up.

Based on the card-sort data, I had initially designed a navigational structure in which the user first selected a category of data, and then searched for items in that category. Assuming from the card-sort data that a search for one type of data was independent from searching for another type, I designed the interaction such that searching for something within any category of data was independent of searching for something within any other category of data. That is, if the user searched for and found a case, and then navigated to the category of Documents, or to the category of Media Contact Information, the initial design assumed that the user wanted to start an independent, unrelated new search in that category. However, further discussion with users revealed that in fact when the user searched for and found a case, it was most likely that he or she would then want immediate access to documents related to that case, or contact information related to that case. Similarly, if the user initially searched for a particular reporter's contact information, then he most likely would be interested in then seeing all documents related to this reporter, and/or all cases related to this reporter. This required a very different navigational and interaction design. Even though card-sorting data had been collected, without the ongoing input of users during early design, it would have been very easy to design the application in such as way as to make the users' most typical type of task very tedious and cumbersome.

Clearly, it is possible, even likely, to design a user interface that does not support users, their tasks, or their work environment if an in-depth requirements analysis is not conducted prior to design and incorporated into design. The chapters that follow describe and discuss a variety of techniques for gathering these all-important requirements specifications.

References

Beyer, H., & Holtzblatt, K. (1998). *Contextual design*. San Francisco: Morgan Kaufmann Publishers, Inc.

Coble, J. M., Karat, J., & Kahn, M. G. (1997). Maintaining a focus on user requirements throughout the development of clinical workstation software. *CHI '97 Proceedings.*

Hackos, J. T., & Redish, J. C. (1998). *User and task analysis for interface design*. New York: John Wiley & Sons, Inc.

Mayhew, D. J. (1999). *The usability engineering lifecycle*. San Francisco: Morgan Kaufmann Publishers.

Pruitt, J., & Adlin, T. (2006). *The persona lifecycle: Keeping people in mind throughout product design.* San Francisco: Morgan Kaufmann Publishers/Elsevier.

Rubinstein, R., & Hersh, H. (1984). *The human factor: Designing computer systems for people*. Burlington, MA: Digital Press.

·48·

TASK ANALYSIS

Catherine Courage
Salesforce.com

Janice (Ginny) Redish
Redish & Associates, Inc.

Dennis Wixon
Microsoft

Introduction **928**
Defining Task Analysis **928**
Task Analysis in this Chapter 928
Considering Four Principles That Underlie Task Analysis 929
Task Analysis Is an Integral Part of a Broader Analysis 929
Task Analysis Includes Understanding Users' Goals 929
Considering Norman's entire action cycle as task analysis 929
Task Analysis Is Relevant at All Stages of the Process 930
Practical Reality Impinges on What We Actually Do 930
Selling Task Analysis **930**
Dispelling the Myths That Lead to Resistance 931
We do market research; We do not need task analysis. 931
Task analysis is too time consuming and costly 931
Using Other Opportunities to Sell Task Analysis 931
Making Task Analysis Part of the Standard Development Process 931
Planning For a Task Analysis (Issues to Consider) **931**
Background Research 931
Getting Into the Project Plan 932
Getting Sign-off and Meeting Schedules 932
Providing Useful and Usable Data 932
Deciding What the Project Team Needs 932
Where is the product in its overall life cycle? 933
How much time do you have to conduct your task analysis? 933
How broad or specialized is the user population for the product? 933
How widespread geographically, culturally, and linguistically is the user population for the product? 933
How detailed must we specify the tasks? 933
Is this a special type of product for which traditional task analysis may not be useful? 933
Deciding on an Appropriate Level of Granularity 933
Deciding Where to Start 933
Gathering Reusable Data 933
Going to Different Users at Different Times 934
Collecting Task Analysis Data **934**

Preparing Philosophically to Work with Users 934
Treat the user with respect 934
Understand that you are a visitor (guest) in the users' environment 934
Be in learning mode 934
Let the user be a mentor 934
Be a partner in the task of understanding the user's world 934
Selecting Users and Environments 934
Make a convenience sample as representative as possible 934
Do not limit task analysis to development partners 935
Conducting Site Visits 935
Observing Users 935
Interviewing Users 936
Doing Rapid Task Analysis 936
Narrow the scope 936
Using multiple teams 936
Schedule your sessions around activity peaks 937
Bringing Users to You (Field Studies in the Lab) 937
Using the lab for procedural task analysis 937
Using the lab for quantitative task analysis 937
Conducting Remote Task Analysis 937
Conducting International Task Analysis Studies 938
Selling an International Study 938
Preparing For Your International Task Analysis 938
Translators 938
Recruiting 938
Logistics 938
Background research 939
Doing rapid international task analysis 939
Hire local consultants 939
Conduct the study in your language 939
Collecting the International Task Analysis Data 939
Being culturally sensitive 939
Being aware of elements that differ in different countries and cultures 939
Analyzing and Presenting the Data 940
Analyzing the Data 940
Involve the design team 940
Make it traceable 940
Make it visible and accessible 940
Match the form to the questions, the stage, and the team's needs 940
Presenting the Data 940
Affinity diagrams 940
Artifacts 941
Flow diagrams 941
Personas 941
Scenarios 941
Sequence diagrams 943
Tables 943
User needs tables 943
Mood Boards for quick data dissemination 943
Live from the field: Another technique for quick data dissemination 944
Culture cards 944
Culture Scapes 946
Culture capsules 946
Conclusion 946
References 946

INTRODUCTION

Successful design comes from a marriage of users' goals and (usually) new technologies. Successful design does not necessarily perpetuate users' current ways of working, but it is built on a deep understanding of those ways and of how a new design will change them.

In this chapter, we explore modern interpretations and uses of task analysis. The first section defines the meaning of task analysis in this chapter.

The five following sections serve as a practical guide.

DEFINING TASK ANALYSIS

Task analysis has different meanings to different authors. Rather than delve into all of the different interpretations, we give a single definition here. (For a thorough comparison of task analysis definitions, see Redish & Wixon, 2003.)

Task Analysis in This Chapter

Task analysis means understanding users' work. Thus, task analysis encompasses all sorts of techniques, including naturalistic observations, interviews, shadowing users, doing day in the life of studies, conducting ethnographic interviews, and observing and listening to users doing specific tasks. It includes gathering information that leads to insights about users' lives at work, to scenarios and use cases, and sometimes to detailed flowcharts of work processes or specific procedures.

A major emphasis of task analysis is predesign, and three types of analysis—user, task, and environmental—are necessary input to designing any product. Therefore, task analysis is an integral part of a triangle that covers users, tasks, and environments. As

described in more detail in the following sections, task analysis goes hand in hand with understanding users (user analysis) and understanding the users' physical, technological, cultural, social, and political environments (environmental analysis).

Users are critical to all three types of analysis. Task analysis requires watching, listening to, and talking with users. Other people, such as managers and supervisors, and other information sources, such as print or online documentation, are only secondarily useful for task analysis. Relying on them may lead to a false understanding.

We believe that task analysis does not stop with design (cf., Kirwan & Ainsworth, 1992). At every stage of the design and development process, task analysis is critical. It is the major input to use cases and design specifications, and it helps us understand how the emerging product affects users. It is the key to evaluating designs as scenarios for heuristic evaluations and for usability testing. Task analysis must be the organizing principle for documentation and training.

We recognize that efficiency oriented, detailed task analyses, such as TAG and GOMS, have a place in evaluating some products, especially those for which efficiency on the order of seconds saved is important (for example, see Gray, John, & Atwood, 1993). However, that type of task analysis is not the focus of this chapter. The focus here is a broad understanding of the world in which the new product will be used.

Considering Four Principles That Underlie Task Analysis

The practical advice for task analysis later in this chapter is based on four principles:

1. Task analysis is an integral part of a broader analysis that includes understanding users and their environments.
2. Task analysis includes understanding users' goals.
3. Although the focus, methods, granularity, and presentation of information may differ at different times, task analysis is relevant at all stages of the design and development process.
4. The practical reality is that task analysis for a given project depends on many factors.

Task Analysis Is an Integral Part of a Broader Analysis

The first principle is that task analysis by itself is not enough to give you the understanding that you need to design or evaluate a product. The methodology you need brings together information about three interwoven elements: users, tasks, and environments.

Fig. 48.1 shows how all three of these analyses must come together as you triangulate what you learn to gain the understanding you need.

Task analysis is one corner of the triangle of understanding that you need. The other two are

- **Users**
  - Who are they?
  - What characteristics are relevant to what you are designing?
  - What do they know about the technology?
  - What do they know about the domain?

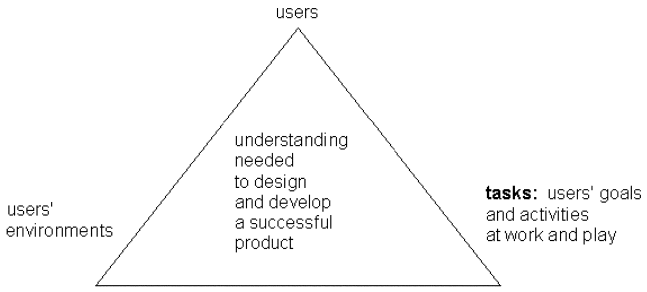


FIGURE 48.1. Triangulating information about users, tasks, and users' environments.

  - How motivated are they?
  - What mental models do they have of the activities your product covers?

- **Users' environments**
  - Physical situation in which the work occurs
  - Technology available to the user (What you might need to find out about could range from what modem speed users have to how often power is interrupted to the cost of upgrading equipment to finding an opportunity for your product because the users' environments favor low technology.)
  - Social, cultural, and linguistic considerations (What will make the new product acceptable in the users' world? How will the new product change the users' world? How will you help the users make the transition from the old world to the new world?)

Task Analysis Includes Understanding Users' Goals

The second principle is that a task is what someone does to achieve a goal.

Considering Norman's entire action cycle as task analysis. As Norman (1988) explained, "To get something done, you have to start with some notion of what is wanted—the goal that is to be achieved" (p. 46). It is true, as Norman also said, that we cannot always articulate our goals clearly. However, in general, we do start with goals, such as

- Making the family happy by getting dinner on the table
- Getting a draft of a paper to a coauthor in a different city

Norman (1988) also described how we go about trying to meet our goals, or how we act. He gives us the seven-stage cycle shown in Fig. 48.2.

We can think of forming the intention as deciding what tasks to do to meet a goal. We can think of specifying an action as figuring out what steps and decisions to take to carry out the task. Executing the action would then be actually doing the steps and carrying out the decisions that we have decided are the way to complete the task.

We can think of perceiving the state of the world as seeing the results of the steps and decisions that we took and inter-



FIGURE 48.2. Donald Norman's view of how people get things done (1988, 48).

preting the state of the world as relating the results to our understanding of how to carry out the task. Evaluating the outcome would then be judging whether we have successfully met the original goal.

Task Analysis Is Relevant at All Stages of the Process

Our third principle is that task analysis belongs everywhere in the process of planning, designing, developing, and evaluating a product. Task analysis, like so much else in the user-centered design process, should be done iteratively. The focus, methods, granularity, and presentation may change over time as different questions and different types and levels of information become more or less relevant (for a detailed discussion of task analysis at different stages, see Redish & Wixon, 2003).

Table 48.1 shows the types of questions that task analysis might help answer at different times in the product life cycle.

Practical Reality Impinges on What We Actually Do

In the fast-paced world of software and web design, in reality, what we can do for a task analysis (or any other aspect of user-centered design) depends on many factors. These factors include:

- Time
- Resources
- People
- Availability of users to observe and talk
- Travel restrictions

This chapter is meant to help you consider and decide the best approaches for whatever situation in which you find yourself. First, we discuss ways to convince clients and others in your company of the importance of usability techniques like task analysis.

TABLE 48.1. Task Analysis Questions at Different Times

Stage	Examples of Questions That Task Analysis Should be Used to Answer
Strategic Planning	Why would someone or some organization choose to use this product? What goals in their world would this product help to meet? What benefits are most meaningful and valuable to users? How will this product be perceived by different cultures?
Predesign	What are the alternatives currently available and technologically possible that would address the why questions listed above? How do users achieve relevant goals today? What works well and what does not? How could our product make that easier?
Information Architecture	What do users know and what are their environments? How do users organize their world? What vocabulary do users use today for their goals and tasks? How can we incorporate that vocabulary?
Concept Design	What metaphors are familiar to users?
Interface Design	What do users know about interface conventions? How does the task flow of the new product match users' expectations from their current work? If we are changing users' task flows, how can we build in help for transitions?
Early Prototypes	What tasks should we provide to heuristic reviewers and usability testers?
Development	What does the user know that would address the problems we have uncovered? What changes should we make to the interface and information to better match users' expectations and work?
Post-release	How well does this release match the user/business needs that we uncovered in the original strategic planning phase? Are users now better able to achieve their goals than they were before they had this product?

SELLING TASK ANALYSIS

In the field of usability and design, usability testing has become a well-understood part of the user-centered design process. This is a great advance in product development. However, equally important upfront activities, such as task analysis, have not yet made the same strides.

Usability testing is more easily justified and understood because it is tied directly to the product or a prototype of the product and it often produces quantifiable metrics. In addition, numerous published case studies demonstrate the value of usability testing.

Even product teams that are very receptive to usability testing may be resistant to task analysis and other requirements activities. You may need to be prepared with evidence of the merits of task analysis and other upfront activities. Selling task analysis is critical; it does not matter how well you understand how to do an activity if you are unable to conduct it.

Dispelling the Myths That Lead to Resistance

Resistance comes primarily from a lack of understanding. Team members may respond to requests to do task analysis with statements like these:

- We do market research; we do not need task analysis
- Task analysis is too time consuming and costly

We do market research; We do not need task analysis. A product team may say, "Marketing has talked to the users of our product." This may be true, but the goals of market research and task analysis differ. Marketing focuses on business decisions and forecasting. They want to understand what will make the product sell. User research, including task analysis, focuses on understanding how the product will be used, so that the product can be designed in a way that is useful and easy to use for the user.

Market research tends to focus on breadth, while task analysis focuses on depth. For example, while market research may tell you that users are interested in a particular feature, task analysis tells you how they would use the feature.

It is also important to note that market research may not always collect data from true end users. In many cases, the person who buys the product (marketing's target) is not the same as the person who uses it (task analysis' target). A good example is software for running a call center. The CIO or senior level management drives the decision as to what call center software to buy, but they will never use it.

The two types of research differ. Both are equally important and have their place in product development. They can even be very complementary. However, their goals, and thus their methods, differ.

Task analysis is too time consuming and costly. Time and resources are two key restraints of product development. Product teams are often hesitant to invest in activities that will impact time and resources if they are uncertain of their true value.

However, the truth is that the return on investment of task analysis is high. Task analysis may cost time and money upfront, but it will ultimately save time and money, and even cause your company to earn more money in the end.

Task analysis saves money during design. Understanding how users work makes the design stage move much more quickly. You also avoid the many design bottlenecks that come from each team member feeling that his or her viewpoint represents the users' viewpoints.

In addition, task analysis saves money during development. User requirements can help to keep feature creep in check. Removing this disruption for development can be a huge time and money saver.

Task analysis also saves money during usability testing. You will find that your product tests better and that fewer design iterations are required.

Because a product that meets users' needs decreases support calls and training costs, task analysis saves money after sales. This is a benefit for your company and the companies and individuals who buy your product.

Task analysis can increase your company's profits. Customer satisfaction is high when products are easy to use. Giving users a feature that they cannot use has no value. Task analysis can create loyal return customers who learn to expect a satisfying experience from your products.

Furthermore, not all task analysis is time and resource intensive. People sometimes equate task analysis with true ethnographic research, which can take months to collect and analyze data. However, most task analysis is not done that way today. Task analysis methodologies have adapted to product cycles and to the desire to be the first to market. Some types of task analysis activities can be planned, carried out, and analyzed in a week. (We discuss these rapid methods later in this chapter.)

Using Other Opportunities to Sell Task Analysis

If your explanations and attempts at education seem to fall on deaf ears, do not despair. You will have plenty of opportunities to raise the topic again because task analysis is relevant throughout design and development. For example, if your product team is spinning wheels and having difficulty making design decisions, suggest doing task analysis again.

In the worst-case scenario, where you just cannot convince the team to incorporate task analysis, they may learn the lesson the hard way—in the usability lab. A product that was developed without the benefits of user research will often fail in usability testing. The team may realize at this point that task analysis might have prevented the failure, and they will likely reconsider next time.

Making Task Analysis Part of the Standard Development Process

Ideally, you want to make task analysis a part of your standard product development process. There are many ways to do this.

- Once you have some data, spread the word.
- Advertise your success stories.
- Hold brown bags to demonstrate the success of your research.
- Post visual posters of your results in hallways where those passing by can see your work.
- Identify leaders in your organization and make them aware of the impacts of task analysis. They can provide top-down influence to get other teams onboard. Soon teams will be coming to you looking for your support.

PLANNING FOR A TASK ANALYSIS (ISSUES TO CONSIDER)

Background Research

If there is an existing product, learn as much as you can about that product before you begin your task analysis. One of the

best ways to do this is to use the product. You may not be a true end user, but becoming familiar with the user interface and the process is beneficial. When you are observing users, you will have a level of familiarity with the product, which can allow the session run with fewer interruptions.

Server log files can be a very valuable initial source of task information. They are a record of file requests to a server and the server responses to those requests. Information from these logs can help determine what you want to observe in your task analysis. From these records, you can learn about

- Visitors—Who is coming to your site and when they return?
- Click paths—Where do users enter your website? How are they navigating from page to page? What is the sequence of the pages visited?
- Time per page—Where was time spent across the pages visited? Where are users spending the bulk (or least) of their time?
- Exit point—Where are users leaving the website?
- Actions completed—Are users completing tasks (e.g., make a purchase, download software, etc.).

There are a variety of other sources to gather information from. These include the product team, marketing, customer support cases, early adopter feedback, and competitor products (for a detailed discussion, see Courage & Baxter, 2004, chapter 2).

Getting Into the Project Plan

A critical aspect of being able to do task analysis throughout a project is getting this and all other usability activities into the project plan. The extent to which this can happen depends, of course, on buy in from the project team, but also on other factors, such as whether a project plan exists and in how much detail it is specified. The more strongly a project team uses a formal project plan, the more critical it is to get usability activities, such as task analysis, into the plan. Time, resources, and respect from managers and developers for the information may be dependent on being part of the formal plan.

Another approach that many usability specialists follow is to create a usability project plan that parallels the system design and development project plan. That is fine if the system people understand and respect the parallelism of the plans.

Getting Sign-off and Meeting Schedules

Whether the usability plan is part of the overall project plan or of a parallel track, it is important to get sign-off from the rest of the design and development team with respect to

- Activities that the usability team will do
- Resources needed for those activities
- Information that will be brought back from those activities
- Deliverables (formal or informal) that will come from that information
- Dates for those deliverables

As usability specialists, we must acknowledge that, at least in the United States, most software and web development projects are schedule-driven. Task analysis and other usability activities must deliver information in a timely fashion at the right moments in the project schedule.

Getting into the project plan early can help set schedules that allow time and resources to do task analysis and other usability activities. Another way that some usability specialists accomplish this is to take responsibility for elements of the project plan, such as the user interface specification, and use that as the way to integrate usability data into product design and development. A project plan is an excellent tool, but we are aware the project schedules often change. You will be dependent on others for your activities, so it is important to track the schedule of activities that you may not be a part of, but that may impact your schedule. Flexibility in your project plan is critical. In some cases, you may find that you have two extra weeks, but more likely, you will find that you have lost time and you need to adjust your usability plan.

Providing Useful and Usable Data

As usability specialists, we help project teams only if we provide useful and usable data when they need it. Therefore, as noted, staying closely aligned with a project plan and schedule is critical.

What else, besides timeliness, makes data useful and usable?

- Data the project team needs. As usability specialists, we should approach any project looking for where the team needs data about users (whether they know it or not) and plan to collect, interpret, and present that data in a way that the team can understand and directly use. Often, the challenge is getting the team to allow you to collect the data, but in other instances, they may want to collect too much data. They may have questions that are much too specific given the stage of development. This is fine if the data can be used later, but if there is no rationale for how the data will impact the product (now or later), it is best to invest your time elsewhere.
- Data that is credible. As we have said, time pressures and limited resources often curtail the extent of any usability activity, including task analysis. In general, we follow the maxim that some data about users and their work is better than no data. Practical experience tells us that is true—as long as the data is accurate and representative. Limited data can still be good data. Good data is collected in a systematic, careful, rigorous way from appropriate users. To do this successfully, you must become familiar with the product and domain. This is critical for credible data collection. Domain familiarity will allow you to identify the appropriate type of task analysis and to formulate the most appropriate questions and observations. The team will also take you more seriously and view you as a true member of the team if you understand the domain.

Deciding What the Project Team Needs

The task analysis that you want to do will depend in part on the type of product or website that you are working on. Consider

these six factors as you think about the project for which you are planning task analyses and other usability activities.

Where is the product in its overall life cycle? For example, is the team

- Upgrading an existing product without changing the medium (a new software release or a revision of a website)?
- Changing business processes or medium (going from a legacy—DOS-based or green-screen product—to a GUI or to the web)?
- Developing something totally new?

How much time do you have to conduct your task analysis?

- Has the team always been onboard with task analysis? Did they defer to you to determine the appropriate activity and time line?
- Has the team suddenly discovered that they need some data quickly and they have not allotted for it in their schedule?
- Does the team believe that there no time in the schedule for task analysis?

How broad or specialized is the user population for the product? Is the product for a

- Very broad public market?
- Niche business market where you can easily define the user population and access to them is through account executives (or similar)?
- Special audience (e.g., children, the elderly, or persons with disabilities)?

How widespread geographically, culturally, and linguistically is the user population for the product? Is the product

- Global? How far? How many countries? Cultures? Languages?
- Local? (Some products are still used only in a particular country—but even within one country, you are likely to find differences in culture and vocabulary.)

How detailed must we specify the tasks? Is the product

- Safety-critical, where tasks are very specific and must be done in specified ways? Do users receive training until they prove their competence in completing the tasks accurately and efficiently?
- Used by many different types of people for different tasks, which they may do in different ways?

Is this a special type of product for which traditional task analysis may not be useful? Some applications do not fit the traditional approach for task analysis. These are primarily applications that the user does for fun, such as games.

Traditional task analysis would not aid much in the design and development of these applications.

As a result, game designers do not typically do task analysis. They are more interested in mood, theme, story, drama, progression, surprise, pacing, and the physical correlates of these experiences. The conceptual space in which they operate and their ways of thinking about how to create a fun experience is fundamentally different, perhaps even antithetical to, the analytic approach that is typical of task analysis.

Deciding on an Appropriate Level of Granularity

Another aspect to consider as you plan a task analysis is the types of analysis to do. Understanding users' goals and their work can be done at several different levels. You might be interested in one or more of these types of analysis:

- **Analysis of a person's typical day or week** (A day in the life of or an evening at home with)—This is probably most needed for the early stages of strategic planning and predesign.
- **Job analysis** (all the goals and tasks that someone does in a specific role—daily, monthly, or over longer periods)—Again, this is probably most needed for the early stages of strategic planning and predesign.
- **Workflow analysis** (process analysis, cross-user analysis, how work moves from person to person)—This may be useful in strategic planning and predesign, especially if you have ways of improving workflow among users. It is also needed in concept design and interface design.
- **High-level task analysis** (the work needed to accomplish a large goal broken down into subgoals and major tasks)—This is needed in information architecture, concept design, and interface design.
- **Procedural analysis** (the specific steps and decisions the user takes to accomplish a task)—This is needed at the interface design stage and beyond.

Deciding Where to Start

You must first understand how far along in the process the project is. Unfortunately, by the time usability specialists know about the project, the strategic planning and predesign stages may already be considered closed. The project may be at considerable risk if the strategic planning and predesign questions were never answered or were answered based on speculation or internal discussions without users. However, it may be unproductive to spend time and effort collecting data that speaks to those questions if no one is willing to listen to the answers you bring back. If that is the case, a more productive use of the limited time and resources for task analysis and other usability activities would be to understand where the project is and how to influence it from that point forward. In some cases, the team realizes (often with your help) that they do need to take a step back to the earlier stages and collect user data in order to prevent the product from going down the wrong path.

Gathering Reusable Data

Time, resources, and costs are likely to limit the number of times you can return to users for task analysis. In addition, users

do not think in terms of the information needs of a project team or the different stages of design and development of products they might use in the future.

One good approach might be to collect extensive data about users' work in a relatively holistic way, such as capturing that data on video or audio or in extensive notes so that you can return to the data—rather than to the users—with different questions in mind at different times.

To do this, an open-ended field study method combined with detailed information gathering is best (Wixon, 1995). Also, having a relatively detailed log of the raw data is necessary (videotape, audio tape, or verbatim transcripts).

Going to Different Users at Different Times

Although the number of times you may go out to users is likely to be limited, we have also said that you are likely to want to do site visits at different times for different questions. Even if you gather extensive data holistically on early site visits and return to the data, you may not have the information that you need. In that case, go out again, and go to different users.

You can use each set of site visits not only to answer the specific immediate issues and questions but also to enrich the team's general understanding of users, their work, and their environments. Although your immediate focus may be a specific why or what or how question, always drill down so that you are, in fact, seeing the why behind the what or the what and how behind the why.

COLLECTING TASK ANALYSIS DATA

While traditional task analysis focused on time and motion studies, modern task analysis relies more on ethnography and cognitive psychology, and that is the focus we take in this chapter. (For a more detailed discussion of task analysis roots in ethnography and cognitive psychology, see Redish & Wixon, 2003.)

Preparing Philosophically to Work with Users

From ethnography and cognitive psychology, we also take our ideas of how to work with users when we are doing a task analysis. Anyone who works with users to gather data for a task analysis should abide by these five principles (inspired by and adapted from the work of Wixon, Holtzblatt, & Knox, 1990; Whiteside, Bennett, & Holtzblatt, 1988; Wixon & Jones, 1996; Wixon & Ramey, 1996; Holtzblatt & Beyer 1993; Beyer & Holtzblatt, 1995).

Treat the user with respect. You are there to understand who users are, what they do, how they do it, and what they value. No matter how differently they do their work from what you expected, you must respect them, their cultures, their actions, and their decisions. Inform users that they are the experts and that you are there to learn from them. In a working environment, you must emphasize to the user that you are there to learn how they work so that you can make their job easier. Users may assume that you are there to evaluate them or to develop a product that will replace their jobs. You must address concerns like these upfront so that the experience is comfortable, rather than threatening, to the users.

Understand that you are a visitor (guest) in the users' environment. You are there to learn about their realities, not to bring your reality to them.

Be in learning mode. You are not there to train, teach, demonstrate, or sell a new product. This part is often difficult for engineers, training specialists, marketing specialists, and others when they first start doing site visits. They are used to going to customers for these other reasons. Before going out on a task-analysis site visit, they must understand the difference in the reasons for the visit, and they must become comfortable with the appropriate frame of mind and behavior.

Let the user be a mentor. If you are there to learn, you must watch and listen more than you act or talk. Especially in work contexts, one way to do this is to suggest to the user a relationship of mentor and trainee—user as mentor, site visit observer as trainee.

Users are often unsure of how to deal with the strangers in their midst. They, too, have more often been visited by people training them, demonstrating something to them, or marketing something to them. Helping them to get into an appropriate relationship is useful, and most users are happy to become the trainer or demonstrator.

Be a partner in the task of understanding the user's world. Use a site visit to test out your assumptions. Clarify your understanding of what the user is doing and saying as you go along. Capture the users' words. Do not translate as you take notes.

Selecting Users and Environments

Companies have sometimes commissioned large-scale task analyses as part of their long-term strategic planning; that is, outside of the development cycle of a particular product or version (e.g., Redish & James, 1996; Dray & Mrazek, 1996). That is terrific when it happens.

However, most task analyses today are small-scale studies, especially when they are part of a product-oriented project. The rapid pace of design and development often does not leave time for site visits to more than a few places. The scale is similar to that used in iterative usability testing—six to eight users per study. As with usability testing, it is better to do a few site visits each time for different purposes at different times in the project than to do many up front and then not have resources to go out for other reasons later. Also, as with iterative usability testing, even though each study is small, over the course of the project, you may see many different sites and users (for details on how to recruit the right users, see Courage & Baxter, 2004).

Make a convenience sample as representative as possible. A task analysis study usually uses a convenience

sample. However, care should be taken to make the sample as representative as possible within the constraints of time and resources (see Hackos & Redish, 1998, especially chapters 2 and 7 on defining your users and then selecting appropriate ones for a user and task analysis).

Because it is almost impossible to find one user who truly represents the entire spectrum of users, spending some time with each of a few users is better than spending all the allotted time with just one user. Characteristics on which you might want to base representation include

- Size of the user's company (which may affect task specialization among users, amount of support available to users, technology available to users, etc.)
- Experience in the domain and with the medium being contemplated for the product
- Gender, age, and background (which may correlate with motivation to learn new ways of working and with interests and values related to technology)
- Location and culture if this is a product for a global market

Do not limit task analysis to development partners. Many companies today have development partners who agree to work with them throughout the project. Development partners are usually major customers who get to influence the new product in exchange for allowing their people to participate in customer activities and beta tests.

Working with development partners is a great idea. However, in many cases, development partners are at the tail end of the distributions for both size and sophistication within the company's market. What works for them may not work well for the company's many other customers.

If the project you are working on has development partners, but the company also wants the product to be used in many smaller and less sophisticated places, push to include others in usability activities like user and task analysis. In addition to making project managers realize how unrepresentative users in the development partners may be, you can also often make a cost-benefit argument based on plans for implementing the product in different environments.

Conducting Site Visits

What happens when you actually meet the users? How should you act when working with users?

The two main techniques for collecting task analysis data are observations and interviews. In a typical site visit, you combine both; and when possible, you do them together—conversing with the user about the work as you observe and listen to the user doing the work.

In certain situations, you cannot converse with the user while the user is working. This may happen when

- The situation is safety-critical (e.g., with air traffic controllers; Means, 1993)
- The user is interacting with a client (e.g., in some situations with retail salespeople or travel agents; Redish & James, 1996)
- The users do not want to be interrupted in their work (e.g., with radiologists; Ramey, Rowberg, & Robinson, 1996)

In those cases, you may be able to talk about the work as soon as the user finishes the task (immediate recall) or at a later time using information from videotape or other artifacts to stimulate recall (cued recall). In situations that depend on the user interacting with a customer, you might have one member of the site-visit team act as customer, thus having the user do a real task but giving you the opportunity to talk with the user during the task (for these and other techniques for task analysis, see Courage & Baxter, 2004; for specifics on observing and interviewing during site visits, see Hackos & Redish, 1998).

Observing users. What you take notes on and the level of detail of your notes depends on the stage the project is at and therefore the questions and issues you are addressing in this particular study.

In each site visit, you may want to spend all of your time with one user, the person responsible for the tasks you are investigating. You may want to spend time with different users at the same site if several people at the site do the same tasks, especially if they have different levels of domain knowledge, product experience, or technical skill.

You may also want to spend time with different users if you are interested in work analysis or process analysis (how work flows across users who do different parts of a process). You may want to spend time with the people who do each part of the process. If you have already watched them individually and drawn flowcharts of what you see as their processes, you may want to bring them together in a conference room to go over your process flow and verify it, discuss it, or ask questions you have about it.

If you want to get details of steps and decisions to flowchart a procedure, you may want to slow the users down and discuss each step and decision, asking questions about other situations and how they might change the steps or decisions and about the frequency of different situations.

In all cases, as you observe and listen, you must remember the five points we made earlier in the section on preparing philosophically to work with users as well as these four:

1. **Be friendly but neutral.**
   Be aware of how you give messages with your body as well as with your words.
2. **Be aware of assumptions that you brought with you.**
   As you watch and listen, you are verifying those assumptions or changing your understanding of the user's reality.
3. **Ask questions to clarify.**
   Restate what you think you heard. Your goal must be to bring back the user's understanding of the work. It is very easy to think that you understand when you are filtering what you see and hear through your own view of the work.
4. **Note the user's words.**
   One goal of a task analysis is to understand the user's vocabulary for objects and actions. Again, it is very easy to translate what you hear into the development team's words without realizing that you are doing it—if you are not attuned to the need to capture the user's words.

Interviewing users. Most of the talking in a task analysis should be a conversation with the user about the work during the work. However, you may want to ask questions that are more general.

In general, interviewing after you have observed is better than before. You will have the context of what you saw and heard as a frame for the other questions. You will likely have a set of predetermined questions to ask the users, as well as questions that arose while in the environment. Having some preplanned questions is a good idea, as that allows you to have some consistent data across all of your participants. This is especially helpful if there are multiple researchers conducting the study.

The three most important guidelines for successful interviewing are:

1. **Ask questions in a neutral manner.**
   Do not lead the user. Allow the user to show you their work and draw your questions from what they show you. You may also give them some broad areas of inquiry or focus (Wixon, Holtzblatt, et al., 1990). Try to put the questions in a behavioral context rather than as a simple matter of like or dislike. If you say, "We are thinking of adding a feature to the product. Would you like that?" you are suggesting that the user say, "Yes, I'd like that." If you say, instead, "If the product had a feature, would you use it?" and follow that up with "When and how would you use it?" you are likely to get a more informative answer.
2. **Keep your questions clear and concise.**
   Avoid lengthy multipart questions or questions with unclear language or with your company or product's jargon. If users do not understand exactly what you are asking, they may misinterpret what you are asking for and respond inappropriately. They may feel awkward and embarrassed because they are unsure for what you are looking (for more details about question writing, see Courage & Baxter, 2004, pp. 262–270).
3. **Listen far more than you speak.**
   You can keep a conversation going with prompts and probes that send the message that you are listening and want the user to talk more (for more on interviewing and listening skills for user and task analysis, see Hackos & Redish, 1998, pp. 279–291).

Whenever possible, make an interview behavioral rather than attitudinal. For example, if you have brought along prototype screens of the new product to show as part of your site visit, do not just show them and ask what users think. Make that into a miniature usability test, even if it is in a conference room and not at the user's desk. Ask the user to walk through the screens doing a realistic scenario.

With interviews, as with observations, clarify what you are hearing. Restate so that you know you are getting the user's understanding. Try not to filter through your preconceived notions. Do not translate the user's words into your company or product's jargon.

Doing Rapid Task Analysis

As we have said, sometimes, the team needed the data two weeks ago; sometimes, they just do not have the time or resources for a multiweek task analysis. Several rapid data collection methods have emerged for these situations. We discuss three here: narrowing the scope, using multiple teams, and scheduling your sessions around activity peaks.

Note that rapid task analysis is not right for all situations. If your team is designing or redesigning a very complex system, there may be so much to understand that even a condensed method will be time consuming.

Narrow the scope. One way to do rapid task analysis is to reduce the scope of your study. Instead of trying to understand everything about the users and all of their tasks, focus on collecting data that you will be able to use immediately to make decisions. Narrowing the scope is an obvious choice when you have specific issues or questions that need to be resolved. Do not focus on the big picture; focus on those specific issues.

Breadth is great if you can afford it, but data analysis will be costly simply because you have more of it. Searching for gems within page of notes and video may be too time consuming. Instead, focus on actionable data, which results from a defined research question.

Using multiple teams. Another way to do rapid task analysis is to have multiple teams collect the data to shorten the amount of time required. This is a great way to engage the entire project team. It also may be necessary in situations where your study is international and involves different languages and cultures.

Engaging the entire team can have a powerful impact. Other team members get to see the users' environments firsthand, and they then often feel ownership in the study. In addition, multiple views of the same task can provide a rich data set and better understanding of what is happening in the users' environments.

Involving multiple data collectors is ideal if you are working with a team who understands task analysis. You may have conducted an activity for them in the past where they were very receptive to the data. Familiarity with task analysis is very helpful in fielding multiple data collection teams.

If you take the team data-collection approach, you must train the observers. You need to make sure that everyone is on the same page about the types of data that will be observed and that will be collected. You want to do your best to make sure that data is collected in a consistent manner. Templates are ideal in this situation. We recommend these templates:

- A protocol that gives the procedures and steps that everyone is to follow when with the users. A protocol helps to ensure that all the sessions are conducted in a consistent manner. A useful protocol includes information on how to introduce the session to the participant, how long each part of the session should last, how to end the session, and so on.
- A list of questions to ask either during or after the session.
- An observation guide that includes the issues on which observers should focus. This helps guide the observations to ensure that everyone is observing the same types of activities. It is not a specific list of questions to ask the user.
- A template for recording data. This may be in spreadsheet or document format. A template for recording data helps cue

observers as to what behaviors they should be observing and provides a consistent data collection method. It makes analyzing data much easier.

- A template for reporting on the session. This is a standard worksheet that each observer completes at the end of each session to highlight some of the key observations or discoveries made. It can be very useful when the team first regroups to discuss the data. In some cases, teams create templates based on the form of analysis required by the development process. For example, at DEC where Quality Factor Deployment (QFD) was extensively used, teams would prepare a QFD template and fill in the template from daily debriefs. The result was an extensive analysis in a useful form on the day they returned from an extensive visit.

If the data is being collected over a number of days, check in with your team at the end of each day to talk about key observations. This can be in person or over the phone. It is great to discuss these things when the data is fresh. Daily check in also gives you an opportunity to resolve any procedural issues that observers may be having.

Schedule your sessions around activity peaks. If you schedule your visits during times of peak activity, you may observe more activity during a shorter window of time. While this technique is useful in any task analysis, it may be particularly helpful if you are doing rapid task analysis.

Our general recommendation that a team of two go to make observations may be critical if you plan to go during a time of peak activity. In a time of peak activity, it is especially difficult for a single person to keep up with the potentially rapid pace. In our two-person teams, one observer focuses on asking questions, while the other takes notes.

Be aware, however, that many users may not want you to come at times of peak activity precisely because they are so busy and stressed. They may feel that they cannot cope with any distractions or additions to the environment. If seeing times of peak activity is crucial to your issues, you may need to convince users that you will observe unobtrusively and not get in the way. You may need to forego some of the questions and interview time that you would like to have.

Bringing Users to You (Field Studies in the Lab)

The best task analysis, of course, is done in the users' context. However, time constraints, travel restrictions, or security restrictions may make it impossible for you to go to the users. You can still do task analysis. Have the users bring their tasks to you.

Although you will not get a true environmental analysis (seeing the users work in their own settings), you can still get a lot of data about users and tasks. When you do task analysis in the lab, make sure you ask the users to bring artifacts relevant to the task being discussed, such as objects participants use or objects resulting from the tasks (Butler, 1996). Artifacts could be a calendar, a procedure manual, a physical form, a computer-generated report, and so on.

Using the lab for procedural task analysis. For detailed procedural task analysis (getting down the steps and decisions in completing a specific task), the lab may be the right environment. If you want to capture details, you probably want a videotape record. It is easier to videotape in the lab than to cart equipment to users' sites, set it up, and take it down—although we do recommend videotaping for most task analysis site visits.

Procedural task analysis can be done either individually or with a group of users. In an individual task analysis, you work with one user at a time and delve into exactly how that user completes the process. In a group task analysis (GTA), you focus on four to six users together and gather the task flow from the group.

Group task analysis has several advantages, including these three (for more on group task analysis, see also Courage & Baxter, 2004):

- Seeing the details. Describing a task flow can be difficult. It is easy to overlook routine steps or to forget about some of the details. Like focus groups, a GTA generates group discussion and a synergy among the participants. Because the users are comparing how they each complete the task, details tend to emerge in the discussion.
- Coming to agreement. One of the goals of a group task analysis is for the participants to come to agreement on a single task flow that represents a common way that they can work to complete the task. When conducting individual task analysis sessions, the researcher must combine the data and make compromises, rather than having the users make these decisions.
- Saving time. You can often develop a rich task flow with a group of users in 2 hours where you would spend 8 to 10 hours spent with individual users. This makes group task analysis another good technique for rapid task analysis.

Using the lab for quantitative task analysis. If you want quantitative measures, such as time to complete a task under ideal conditions, the lab may again be the best place to do the study. Someone might ask, in that case, whether you are doing a field study or a usability test, but in fact, the line between the two techniques is quite blurry. Both can be done in the field or in the lab, on old products or new prototypes, qualitatively or quantitatively, and on the user's own work or on scenarios given to the user. Task analysis is a major part of both field studies and usability testing.

Conducting Remote Task Analysis

If your focus is software or the web, you can do remote task analysis to collect useful data. You can use a screen-sharing application to see the user's desktop as the user works. As you cannot with lab studies, you cannot observe the user's physical environment. However, one advantage you have over a lab study is that the users are still in their own environments. You can ask users to describe their environments, and some may even be willing to send photos, which can give you useful insights.

Remote task analysis is sometimes your only option, especially in international studies. You may not have the resources

for global travel. These types of situations are best when you already have some experience or familiarity with the culture.

A remote task analysis can also be very useful to supplement onsite sessions. If you have only the resources to visit one or two users, do so; then, supplement your data through remote sessions with other users.

CONDUCTING INTERNATIONAL TASK ANALYSIS STUDIES

We have put international task analysis in a separate section to give special attention to the unique considerations you must attend to at each stage: selling, preparing, and collecting the data. (We will also give special techniques for presenting international data later in the chapter.)

Selling an International Study

Many of the arguments for international task analysis are the same as those for local studies. However, as Siegel and Dray (2005) pointed out, the costs, risks, and benefits of international research are often an order of magnitude greater than for local studies. For example, a product team's assumption that they represent the user is particularity dangerous when the users are from a different country and culture. In addition, if the team assumes that the product's functionality and design are so compelling that users all over the world will adopt it, they are likely to find that is simply not the case. What is compelling in one culture may not be in other cultures.

International research is often most effective and important when the company is taking a strategic-level initiative and going after a market in a new location. That is where many companies are today, particularly those on the Internet, as there is far more growth potential outside North America and Western Europe (Siegel & Dray, 2005). If international adoption is part of the business objectives for the product and will be a factor in determining the product's success—that is, if your company is serious about selling internationally, then they absolutely must be equally serious about understanding the users in those markets.

Many product teams think that if they have a team member from the geographic or cultural area of interest they do not have to do any user research, including any task analysis. This is not true. A representative from a major user group can be very helpful, but Siegel and Dray (2005) gave several reasons why it is not enough. First, products often require specific domain knowledge. If the representative does not have this domain knowledge, the team may be misguided. Second, just because a person was raised in a particular country or culture does not mean that person can speak about the impacts of the culture on software designed for a specific profession within that culture. Third, a person who has been out of the country or culture of interest very likely has lost at least some cultural awareness—and may be out of touch with the current cultural situation. Fourth, the fact that the person relocated may make that person different from those who chose to stay.

Another invalid assumption is that simply translating the product into the local language is enough. Text is only one aspect of an interface. Localization takes the effort a step further. The goal of localization is to ensure that all information is adapted to a particular location or culture (Uren, 1997). This can include such information as currency, date formats, colors, icons, and so on. Task analysis is ideal to get this level of understanding. It helps you understand both similarities and differences.

Preparing For Your International Task Analysis

Preparation is a key element of every successful study. International studies add several significant considerations including translators, recruiting, logistics, and background research. After discussing these four considerations, we give some hints on preparing for a rapid international task analysis.

Translators. If the language of your study participants is not your native language, you will want to employ a translator. This person will need to translate all of your tasks analysis materials before hand and then to translate in real time during the session (simultaneous translation). Simultaneous translators are often known as interpreters, and interpreting is a specialized and expensive skill, as it requires a great deal of concentration and focus over extended periods. You will also need to allow time to train the translator in the goals, objectives, and procedures of your study to ensure that as little as possible get lost in translation. (This is, of course, the ideal approach. We present some time and money saving alternatives in the section on rapid international task analysis a little later in the chapter.)

Recruiting. Recruiting for an international study is not the same as recruiting for a local study. Factors that often differ include appropriate ways to contact users, recruitment and incentive costs, no show rates, the times that people are willing to participate, and holidays. Be sure to account for these differences in your plan. If possible, it is wise to engage a local recruiting company. If your company is international, engage the people in the office of the country you will be visiting to help you find users and to learn about the culture in advance (Siegel & Dray, 2005).

Logistics. If you are visiting multiple locations, time your visits so that you can minimize your travel time and costs. You do not want to fly roundtrip to Hong Kong from New York and then two weeks later fly to Shanghai from New York. Also, be sure to give yourself some time to adjust to new time zones. Task analysis activities require you to be very alert and attentive. You will not be of much value if you are completely exhausted.

Also consider your equipment. Will it work in the country to which you are traveling? Will it be too much with which to travel? If you need to scale back, you should consider this early on. Should you rent the necessary equipment? If yes, from where should you rent it? Thinking through all of the details ahead of time will save headaches as your site visit approaches and while you are on site.

Background research. Do as much background research as you can before you go. This will help to prepare you for some of the things you many encounter. For example, read travel guides and books on cultural etiquette. Read articles in research papers, professional magazines, and on the web. Examine competitor products that may be thriving in the culture you are going to visit. Use this information to guide you as you create your task analysis plan.

This is a lot of preparation to conduct international task analysis. The good news is the preparation is worthwhile. As you come to understand a culture and make connections with people from that location, future studies will be easier and less costly. The understanding that you get from an international task analysis will pay off richly in the product's success in that culture.

Doing rapid international task analysis. At times, time and resources will not allow you to do all the preparation we advise or to conduct an international study in the ideal way. Earlier in the chapter, we discussed conducting remote studies. Two other approaches for those situations are to hire local consultants and to conduct the study in your language. Both approaches are less than ideal and work best when you already have some experience or familiarity with the culture.

Hire local consultants. Having someone local conduct the study can be a tremendous time and resource saver. They can conduct the study in the local language for you and send a translated report. Professional associations, such as Computer-Human Interaction (CHI), Human Factors and Ergonomics Society (HFES), and Usability Professionals' Association (UPA) can be good starting points to look for these people. It may be beneficial to have the data interpreted by someone from the culture of interest and avoid misinterpretations that may occur due to your lack of understanding of the culture.

One disadvantage of this approach is that it can be difficult to find professionals who truly understand task analysis. You want to make sure that they understand the method and goals of your study. A second disadvantage is that seeing is very powerful. A tremendous amount is learned about a culture by being immersed in it. If you do not go, you miss that immersion. A third disadvantage is that if someone who lives in that culture conducts the study, they may take some of its subtleties for granted and not report on them even though they are critical for your team to know. A fourth disadvantage is that if you are conducting research in multiple countries with different consulting teams, it becomes harder to compare and contrast across cultures because different people who may actually never speak with one another collected the data.

Conduct the study in your language. Another way to save time and money is to conduct the study in the language of the product team. This eliminates the need for a translator. It often works best in countries where the team's language is a very common second language. It is particular tempting to do this if English is the team's language because in many countries it can be relatively easy to find English-speaking users.

Much success can result from this method. However, you must consider the fact that people's perception of how fluent they are may not reflect reality. You should always conduct a screening call with these users before visiting them to assess their language skills. Another consideration is that choosing users who speak your language restricts the pool of participants, and as a result, the participants you work with may not be representative of the user population of interest.

Collecting the International Task Analysis Data

As you collect data internationally, you should add two critical considerations to your repertoire: being culturally sensitive and being aware of elements that differ in different countries and cultures.

Being culturally sensitive. Research shows that methods based on the Western thought can lead to cultural conflict and misinterpretation of data, particularly in Asian cultures (Chavan, 2005; Ann, 2004).

For example, Chavan (2005) explained that, in India, users often hesitate to say that something is bad or to identify issues with a product. Even if an Indian user does not highlight any problems, you cannot assume that no problems exist for that user. Users in India may also be more likely than users elsewhere to feel that they are being evaluated just because someone is observing them.

For another example, Ann (2004) said that, in China, relationships are very important. Friendship is a prerequisite to dealing in business or finance. This may make it difficult to recruit participants unless they come from friends' referrals. If you are not connected via a friend, users may not want you in their homes. Even if you are able to enter a home without such a connection, you may not hear the users' true feelings, as they may take great care not to offend you. If you do not continue the friendship after the study, Chinese participants may feel used and decline to participate in future studies. Furthermore, a Western researcher and participant relationship would not be well received in China. The Western approach in which you ask questions in a very scientific manner and reask them in different ways to probe deeper may lead to distrust rather than to useful information. Less intrusive styles may be more appropriate.

Therefore, choosing a culturally sensitive approach to task analysis for the location you will be visiting is critical. This is where local consultants or people in your company at that location can be invaluable. Whether you actually conduct the study or have them do it, they can be extremely valuable in helping your refine your approach so that the experience is pleasant and comfortable for the users and so that you collect valuable and accurate data.

Being aware of elements that differ in different countries and cultures. As Siegel and Dray (2004) recommended, you should be aware of the many elements that change between countries and cultures so that you can collect data on whichever of these are relevant to your product. Here are just a few of the elements to consider:

- Purchasing dynamics and financial transactions (e.g., forms of payment, willingness to disclose financial information to others or online)

- Social structure and service expectations (e.g., society classes, willingness to do it yourself)
- Mental models of geography (e.g., address formats, maps)
- Use of physical space (e.g., size of workspaces)
- Climate and environmental conditions (e.g., how do environmental conditions impact the user of the product)

ANALYZING AND PRESENTING THE DATA

Once you have collected the data, of course, you must analyze it and present it. Data is of no value if you do not communicate what you have learned to the people who need the information. To make the data useful, you must bring the data together, think about what you have learned, and draw out implications.

Analyzing the Data

Consider these four principles as you plan to analyze the data:

Involve the design team. Making the effort to involve the rest of team pays off handsomely. Involving other team members ensures that they have a stake in the results. It also allows them to work with the raw data, which helps them internalize the work of the user more completely, even if they did not get to participate in the site visits. When you involve the team in the analysis, you build a shared understanding of what was seen and heard at the customer or user sites.

It also ensures that the questions the team has get answered. Often, teams refine and redirect their thinking as they go through an analysis. They may drill down more deeply into the data, completely change the questions they have of the data, and may change their thinking about the direction they had planned for the product.

Working as a team can also help you generate actionable items that result from the study. You can work together to prioritize the findings. This is ideal to do as a team because you can discuss limitations, such as time, and determine what is truly feasible.

Make it traceable. Any analysis should include references back to the raw data. There are many advantages. First and foremost, keeping the link ensures the integrity of the analysis. It is important to be able to say to those who were not involved in collecting and analyzing the data that all conclusions are traceable back to statements by users or direct observation of their behaviors. Second, as we noted in discussing the first principle, interpretations may change as analysis progresses. During that process, it is important to be able to revisit the data and recall the context of the behavior or the comments. If you set up ways of tracking the data through analysis, the extra time and effort need not be substantial.

Make it visible and accessible. The analysis may be complex and detailed, but a report laden with text will almost certainly go unread. There is not time in the rush of a project for people to read. Therefore, many teams choose to display their analyses on a wall or in a war room, where team members can review and add comments to the display (e.g., Simpson, 1998.)

An alternative is a hyperlinked document in which higher-level conclusions are linked to more specific analyses. Often, the analysis is graphical so that designers can stand back and see patterns in the data. Some people create multilevel documents with a one-page summary, supported by a 3-page overview, which in turn is supported by a 50-page report.

Match the form to the questions, the stage, and the team's needs. The cardinal rule of all documentation is to give users what they need in the form they need it when they need it. That is why most technical communicators have moved from writing extensive tomes that people do not open to helping teams bring communication into the interface.

The same principle applies to the internal working of any project team. The best form in which to represent the data depends on many factors, including the questions that were asked, the stage in the project's life cycle (that is, how the information will be used), the time in which the information is needed, and the team and company culture.

Keep in mind the essential purpose of any analysis. Analysis provides an anchor from which designs can be generated and against which they can be evaluated. The analysis is not an end in itself; you must keep the design team engaged with the analysis and with the representations of the analysis. In the next section, we describe a few of the possible representations.

Presenting the Data

Here are 13 ways to present task analysis data (see also Hackos & Redish, 1998; Courage & Baxter, 2004, described data analysis for a variety of task analysis techniques; Miles & Huberman, 1994, also described the rich variety of ways to organize qualitative data for interpretation).

Affinity diagrams. Affinity diagrams are hierarchical pictures of user data. They are produced inductively by grouping similar data elements together into categories and then grouping the categories together. Fig. 48.3 is an abstract illustration of an affinity diagram in progress.

Affinity diagrams derive much of their value from the process that produces them, that is, a deep engagement with the data

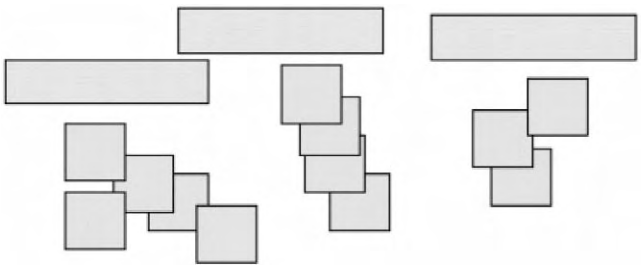


FIGURE 48.3. An abstract illustration of an affinity diagram in progress.

1. **I must** FIND, MAINTAIN, **and** COMMUNICATE **with customers, and sometimes others.**

   1.1. ***I need to keep in touch with clients, colleagues, and family.***

      1.1.1. **I must make sure my landline is forwarded--can't miss calls while I'm out.**

      - P 622 SW 6.2 Seq (Getting ready to leave): Turns off the phone so that the answering machine goes online.
      - P 508 SW 5 Seq NOTE: He has a landline, but has forwarded it to his mobile phone so that he always can get calls when he's away from the office and also so that he doesn't forget to forward the phone when he leaves the office.

      1.1.2. **My cell phone is essential during the business day--it must be within easy reach.**

      - P 349 US 3 Notes He docks his phone on the dash where he can reach it easily.
      - P 442 SW 4 Flow NOTE: The phone is everywhere in this flow diagram—as much as he is hitting the pavement, he is also hitting the wire.
      - P 552 SW 5 Notes If I leave my phone at home, I've ruined my day. I usually have at least 1 visit, and I need the phone [to help if I get lost, to confirm the visit, and to keep track of other business when I'm traveling].
      - P 628 SW 6.1 Seq 6.1 has her bag back there in the trunk too, but her mobile phone is in her hand. [Can't tell if earphone is on; it's not by the time we get out of the car in any case.]
      - P 834 SW 8 Seq In the car he puts the mobile phone between his legs, resting on the seat. BREAKDOWN: I normally have the mobile phone put away in my pocket [or somewhere] and an earpiece in my ear so I don't have to mess with it while I drive. [ASSUME: There's usually a switch next to the hanging microphone to let you answer calls.] But this borrowed phone doesn't have an earpiece.

FIGURE 48.4. A partial example of results from an affinity diagram.

combined with recurring reflections on the generalization that best captures a number of data elements. Teams often produce collateral elements while creating an affinity, such as design ideas and additional questions, that are captured and then used in design or further data gathering.

Fig. 48.4 is an example of a small part of the result of doing an affinity diagram. Note how the team has kept the reference to the data with each bullet point (that is, each note in the affinity diagram).

Artifacts. Some types of tasks are deeply intertwined with their artifacts. For example, the task of making appointments is necessarily interwoven with calendars that show dates and times. As a result, it is often best to begin with existing artifacts (e.g., Ellen's appointment book) and to organize data around that representation.

The team may or may not decide that the representation is in fact the best metaphor to bring into the new product. Even if they do not, the artifacts and representation are often the best way to get an initial understanding for a task analysis. Fig. 48.5 shows an example of an artifact that you might want to collect if you were creating a program to help people keep track of time, dates, meetings, tasks, and so on.

Flow diagrams. Flow diagrams answer questions about how information or artifacts flow through a system (process analysis). They illustrate the dependency between system elements or states of the system and what needs to transferred or moved from one part to another. They also show how roles are divided within an organization as data moves from one person or department to another or between the organization and outsiders.

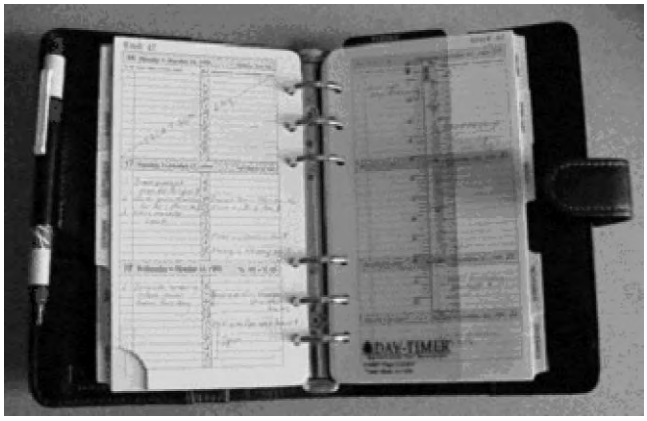


FIGURE 48.5. An example of an artifact.

Fig. 48.6 is a small part of a flow diagram. The different patterns on the underlying circles represent different people (managers, nonmanagerial professionals, consumers, etc.) who are involved in this flow. Note once again that the references to the raw data are kept with the flow diagram so that the team can return to the data to understand more about the user and the context for each data point.

Personas. Personas are composite archetypes that represent the primary and secondary users of the product. A persona description often describes the user's activities, knowledge, and tasks in some depth. Thus, personas may integrate the two elements of users and tasks from the triangle presented in Fig. 48.1. In fact, some people also include information about the persona's environments (physical, social, cultural, technological), thus capturing all the triangulated data that we discussed at the beginning of the chapter.

Rich persona descriptions that encompass user, task, and environment information are particularly useful for commercial products, which often begin with market segmentation that classifies and describes potential customers. The task analysis builds on this data by characterizing these users more precisely (Lee & Mikkelson; 2000).

Because personas instantiate users as actual people, they tend to be memorable. You can ask "How would Julie do her work if we design the product this way?" (see Cooper, 1999). Personas can come to team meetings as life-size cardboard figures (Tahir, 1996), on posters, or on placemats. They can become part of the product team's e-mail group with someone in charge of representing them (Pruitt & Adlin, 2005).

Fig. 48.7 is one of many ways of showing a persona. The real example would include photographs or sketches where indicated.

Scenarios. A scenario is a short story of a specific situation that is real and relevant to a user. A scenario gives the team the user's goal and specific needs. It often also gives the team the user's names for objects and attributes of those objects. It may give the team information on what the user values (e.g., Is price more important than choice in renting a car?).

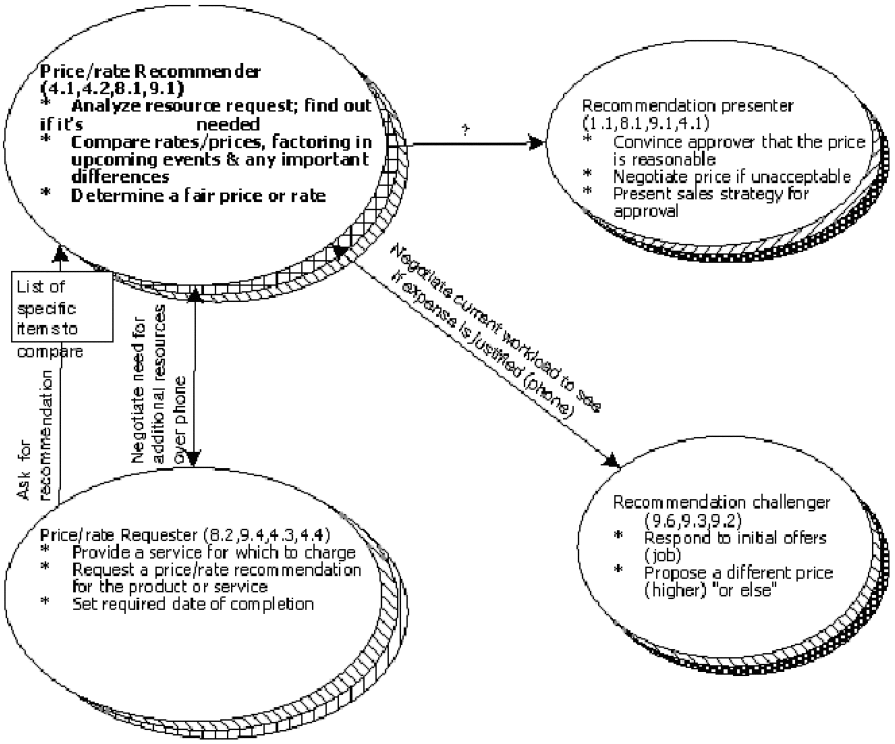


FIGURE 48.6. Part of a flow diagram to present task analysis data.

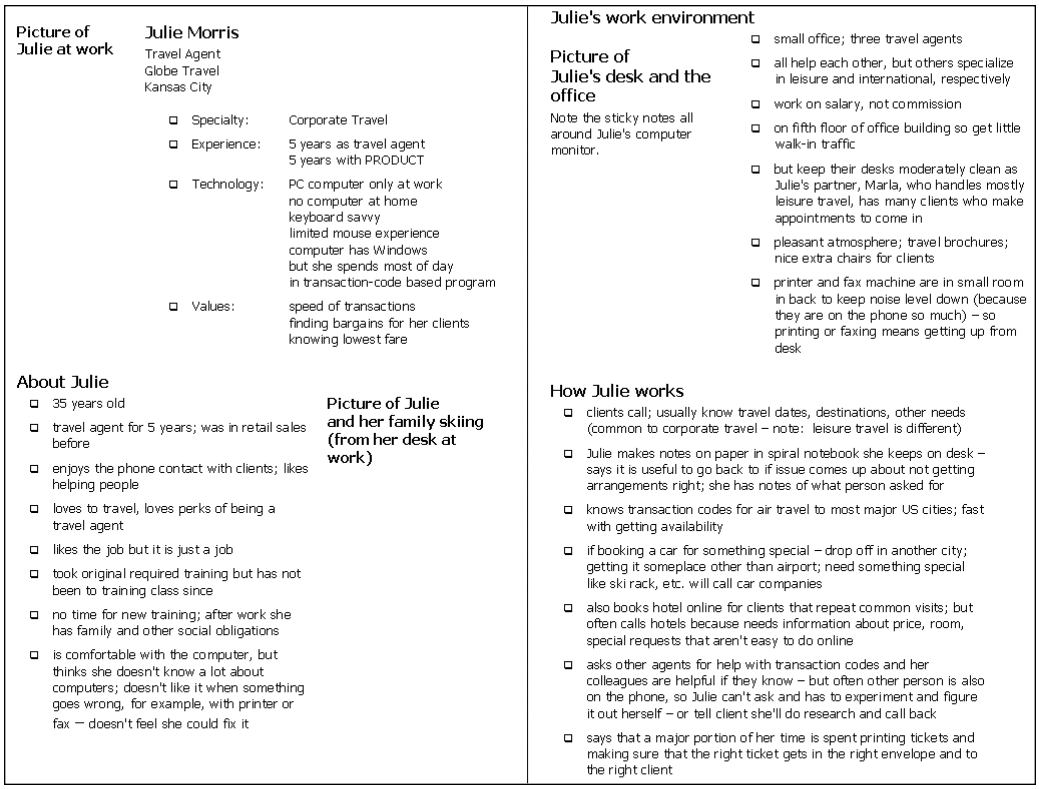


FIGURE 48.7. An example of one type of persona (user profile).

Client calls the travel agent. Client wants to go to Phoenix for a weekend to see her friend, but she is very concerned about the price. She would like to go sometime this month. She can be flexible about when to go and come back. She wants to know what the best fare is, if it is available this month, and what flights she can get for that fare.

FIGURE 48.8. An example of a scenario.

Each situation that you observe on a site visit is a scenario. You can also collect scenarios by interviewing users through the critical incident technique (Flanagan, 1954) in which you ask users to recall a specific incident and then to tell you about it.

Fig. 48.8 is an example of a scenario from a study of travel agents (Redish & James, 1996).

You can elaborate a scenario with the sequence diagram (flowchart) of the procedure the user went through to accomplish the scenario. If accomplishing the scenario is difficult, and the scenario is important, creating a more efficient procedure could become a requirement for the new product.

Sequence diagrams. Flow diagrams track work through a system or across people. Sequence diagrams use time to track the actions and decisions that a user takes. Sequence diagrams (procedural analysis) show what users do and when and how they do it. This type and level of information is critical for interface architecture and design because it gives us the functions, objects, and attributes of a system (e.g., menu items and dialog box design) and the navigation for a website.

By laying out the sequence diagram that represents what users do today, you can often see ways to make the product help users be more efficient and effective. Thus, the sequence diagram of the reality of what you find in predesign observations is often elaborate and messy. It may be important for the team to see that reality as they work towards a more useful product.

Fig. 48.9 is an example of part of a sequence diagram.

Tables. Tables are an excellent way to show comparisons and so are useful for presenting many types of analysis. Technical communicators, for example, have traditionally used a user/task matrix to understand which tasks are done by which types of users. The user/task matrix becomes a major input to a communication plan—to answer the question of what tasks to include in documentation for people in different roles (e.g., system administrators, end users). Tables can be used to show the relationships between any two (or more) classes of data. See Table 48.2 for an example of a table with three classes of data.

User needs tables. Kujala, Kauppinen, and Rekola (2001) developed what they call a "user needs table" as a way to present a current task sequence along with the problems and possibilities that the designers should think about for each step in the sequence. They hypothesized that it would be easier to use findings about users' needs in design if the findings were connected to the task sequence that forms the basis of use cases. In their studies, designers found this type of presentation very useful in moving from data to requirements. They also found that writing use cases from this type of presentation keeps the use case in the user's language and keeps the use case focused

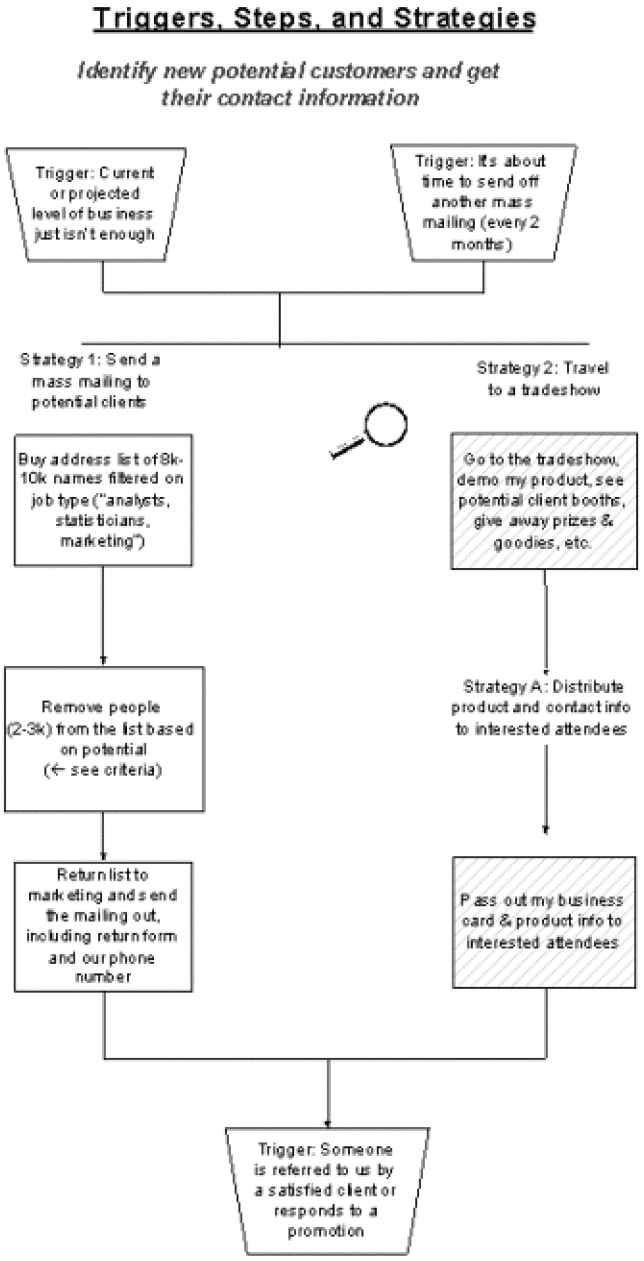


FIGURE 48.9. Part of a sequence diagram.

on the user's point of view. Table 48.3 shows an example of a user needs table.

Mood Boards for quick data dissemination. Mood Boards (Foucault, 2005) are an ideal way to present preliminary research results when your team is under a tight time line and you want to get them some data quickly or when you want to keep the momentum from the study going.

Mood Boards are a collection of raw data and artifacts presented with the goal of inspiring the product team early in the development cycle. They often take the form of a large poster showing photographs, sketches, screenshots, participant quotes, and artifacts from the field, such as forms, printouts, and Post-

TABLE 48.2. Matrix for Weighing Market Size, Users' Tasks, and Potential Product Functions

Users	Relative market size Hi—3 Med—2 Low—1	Hi level task	Task relative importance Hi—3 Med—2 Low—1	Tool Bar Customization	Better drag and drop	Progress meters	Interruption protection recovery	Total (sum of products, how well market is served by these features)
Techno Bob	1	Customize my screen	3	2	1	0	0	9
		Down load information	3	0	0	1	2	9
Newbie Ed	1	Customize my screen	1	−1	2	0	0	1
		Down load information	3	0	2	3	0	12
Practical Sue	3	Customize my screen	2	2	2	1	0	36
		Down load information	2	0	2	1	2	30
Sum				17	35	27	18	
				−1—will confuse users 0—users will not use it 1—users mildly positive 2—users strongly positive				

it notes. You can use Mood Boards to highlight some of the interesting or unexpected findings from the study.

You can put together a Mood Board quickly because the date is not yet refined or synthesized. Mood Boards can also be ideal for international studies because they can convey differences between cultures and can give designers a feeling for the aesthetics of the other culture. Fig. 48.10 is an example of a Mood Board from a study that was part of a project to develop technology for elderly in the United States. Mood Boards were created for areas where there were product possibilities. These included urban elderly, elderly fitness, and fraud against elderly.



FIGURE 48.10. An example of a mood board representing urban elderly. Courtesy of Intel Corporation—Digital Home Group.

Live from the field: Another technique for quick data dissemination. Another very effective rapid communication method is to send information back to the team from the field in real time (Lovejoy & Steele, 2005). This method is particularly useful for international studies where the product team is often quite curious and excited about the study. In addition, if you are going to be traveling for an extended period, you can get some data to the team so they do not have to wait for weeks to hear what you are learning. Photos, stories, snippets of data, and insights can help the team to feel a part of the study.

Lovejoy and Steele (2005), for example, collected photos and narrated the photos with data to create what they call photo stories. Other useful ways to communicate findings quickly include blogs, e-mails, and websites.

Culture cards. The last three data presentation methods focus on conveying the culture of your study participants. These techniques were developed by Intuit (Foucault, 2005) to convey findings from an observational study.

Culture cards are a deck of 20 to 30 physical cards that are used to communicate key findings and images from different cultures. Information such as photographs, brief research findings, user descriptions, or users' needs is printed on materials of cultural significance, such as greeting cards, textbook paper, or handmade paper from the area. Fig. 48.11 shows examples of the types of findings that may be noted on one side of a culture cards. These cards were created for a project on pregnancy and parenting conducted by Intel. On the flip side of such cards, you might find photographs that are representative of the findings.

The intent is to be motivational—to inspire the product tem to create culturally sensitive and appropriate design solutions.



FIGURE 48.11. Findings displayed on the back of the culture cards. Courtesy of Intel Corporation—Digital Home Group.

TABLE 48.3. An Example of a User Needs Table, from Kujala et al. (2001; used with permission)

Task Sequence:	Problems and Possibilities:
Step 1: When trapped in an elevator, passenger makes an emergency alarm.	• Passengers want to get out of the elevator as soon as possible. • All kinds of passengers must be able to make an alarm call (blind, foreigners, etc.). • Sometimes passengers may make false alarms unintentionally. • Passengers may be in panic. • Passengers need instant confirmation that they have created a connection to the service centre operator and that they are going to get help.
Step 2: Unoccupied service centre operator receives the emergency alarm call and asks for information (description of the failure).	• Different versions and types of remote monitoring systems. • Passenger is the only information source. • Service centre operator does not notice the emergency alarm call.
Step 3: Service centre operator completes transmission of information to the system and sends it to the area service person.	• Laborious phase for the service center operator. • Simultaneous calls must be differentiated. • Service person cannot see all the information. • Inadequate information from a site system. • Possibility: Instructions as to how to operate the system. • Possibility: Possibility to open phone line from Call Center to the elevator.
Step 4: Service center operator calls the service person and reads the description of the failure.	• Extra work for the service center operator.



FIGURE 48.12. Photo of a culture scape. Trying to immerse the user in the Asian culture by having the team remove their shoes and by decorating the conference room with artifacts. Courtesy of Intel Corporation—Digital Home Group.

Culture cards can provide a useful overview of key findings and serve as a launch pad for brainstorming and design sessions. Two key advantages are that they break the data into digestible chunks and that they cue the designer to appropriate aesthetics for each culture.

Culture Scapes. When communicating results from an observational study, it is challenging to convey the true feelings and sensations generated by the environment. Smells, sounds, textures, and flavors are all important aspects of the user's environment, but they can be difficult to relay to the team (Foucault, 2005). Images and words are often not enough. Fig. 48.12 is an example of a Culture Scape.

Culture Scapes are designed to allow the team to experience more richly the culture by engaging multiple senses. Foucault (2005) successfully used Culture Scapes when relaying findings from a mobile technology study in Singapore. During the study, the observers were struck by the intense heat and humidity and how it related to their observations. To convey these environmental realities, they presented their findings in a room that was 90 degrees Fahrenheit with 85% humidity. Because of the simulated environment, team members got the importance of these environmental factors and their critical impacts on design. If you plan to use Culture Scapes, it is important while you are in the field to consider what sensory experiences are most important and how you will convey them when you return.

Culture capsules. Culture capsules showcase unfamiliar elements of a culture by physically recreating a space from the observed environment (Foucault, Russell, & Bell, 2004). This presentation method is of particular importance when space is the primary consideration in design. Photographs taken in the field guide these recreations, which may also include artifacts from the physical environment being recreated. For example, Foucault et al. (2004) used this technique to highlight the space constraints in a typical Chinese home so the team would understand the limited space available in a typical home office in China. By replicating the physical spaces they had seen, the researchers made the necessary information about space constraints obvious to the product team. The space also served well as a brainstorming area and a project showcase.

CONCLUSION

Many usability practitioners do task analysis in one form or another, although often they do not recognize it as such, much as Moliere's character found the fact that he had been speaking prose all his life a revelation.

Some may hesitate to plan for task analysis because they think of it as too complex and time consuming to apply in real-world situations where time is always too short. In this chapter, we have tried to show that task analysis is a family of flexible and scalable processes that can fit well in almost any development environment.

Some may have thought of task analysis only in terms of highly structured ways of capturing minute details of specific procedures relevant to evaluating already created designs or developing training or documentation for already determined systems. In this chapter, we have tried to show that while such uses of task analysis continue, the more common use today is in developing a very broad understanding of users' work and is of great use from the earliest strategic planning stages through all the phases of predesign. Task analysis as laid out in this chapter continues to be useful throughout the process, being used at later stages to develop scenarios for user-oriented evaluations as well as in inspection methods.

Task analysis is a way to involve the entire team in understanding users. It provides ways to organize the mountain of unstructured data that often comes from field studies or site visits. It is an essential part of the process of creating any product (software, hardware, website, or document) because products are tools for users to accomplish goals; products are all about doing tasks.

References

Ann, E. (2004). Cultural differences affecting user research methods in China. In C. Courage & K. Baxter (Eds.), *Understanding your users* (pp. 196–207). San Francisco, CA: Morgan Kaufmann.

Beyer, H., & Holtzblatt, K. (1995). Apprenticing with the customer. *Communications of the ACM, 38*(5), 45–52.

Butler, M. B. (1996, January). Getting to know your users: Usability roundtables at Lotus Development. *Interactions, 3*(1), 23–30.

Butler, M. B., & Tahir, M. (1996). Bringing the users' work to us: Usability roundtables of Lotus Development. In D. Wixon & J. Ramey (Eds.), *Field methods casebook for software design* (pp. 249–267). New York: Wiley.

Carroll, J. M., & Rosson, M. B. (1987). The paradox of the active user. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction* (pp. 80–111). Cambridge, MA: MIT Press.

Chavan, A. (2005). Another culture, another method. *Proceedings of the Human Computer Interaction International Conference,* Las Vegas, NV, •••.

Coble, J., Maffitt, J., Orland, M., & Kahn, M. (1996). Using contextual inquiry to discover physicians true needs. In D. Wixon & J. Ramey (Eds.), *Field methods casebook for software design* (pp. 229–248). New York: Wiley.

Cooper, A. (1999). *The inmates are running the asylum*. New York: Macmillian.

Courage, C., & Baxter B. (2004). *Understanding your users.* San Francisco, CA: Morgan Kaufmann.

Diaper, D., & Addison, M. (1992). Task analysis and systems analysis for software development. *Interacting with computers, 4*(1), 124–139.

Dray, S., & Mrazek, D. (1996). A day in the life of a family: An international ethnographic study. In D. Wixon & J. Ramey (Eds.), *Field methods casebook for software design* (pp. 145–156). New York: Wiley.

Flanagan, J. C. (1954). The critical incident technique. *Psychological Bulletin, 51*(4), 327–358.

Foucault, B. (2005). Contextualizing cultures for the commercial world: Techniques for presenting field research in business environments. *Proceedings of the HCII Conference,* Las Vegas, NV, •••.

Foucault, B., Russell R., & Bell, G. (2004). Techniques for researching and redesigning global products in an unstable world: A case study. *Proceedings of CHI2004 Conference on Human Factors in Computing Systems,* Location, •••.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world performance. *Human Computer Interaction, 8,* 237–309.

Hackos, J. T., & Redish, J. C. (1998). *User and task analysis for interface design*. New York: Wiley.

Hackos, J. T., Elser, A., & Hammar, M. (1997). Customer partnering: Data gathering for complex online documentation. *IEEE Transactions on Professional Communication, 40*(2), 102–110.

Holtzblatt, K., & Beyer, H. (1993). Making customer-centered designs work for teams. *Communications of the ACM, 35*(5), 93–103.

Kirwan, B., & Ainsworth, L. K. (1992). *A guide to task analysis.* London: Taylor & Francis.

Kujala, S., Kauppinen, M., & Rekola, S. (2001, December). Bridging the gap between user needs and user requirements. *Proceedings of PC-HCI 2001 Conference*, Patras, Greece, •••.

Lee, W. O., & Mikkelson, N. (2000). Incorporating user archetypes into scenario-based design. *Proceedings of the ninth annual conference UPA 2000*, Chicago, IL, •••

Lovejoy, T., & Steele, N. (2005). Incorporating international field research into software product design. *Proceedings of the Human Computer Interaction International Conference,* Las Vegas, NV, •••.

Means, B. (1993). Cognitive task analysis as a basis for instructional design. In M. Rabinowitz (Ed.), *Cognitive science foundations of instruction* (pp. 97–118). Hillsdale, NJ: Lawrence Erlbaum Associates.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded source book.* New York: Sage.

Norman, D. (1988). *The design of everyday things* (Rev. ed.). New York: Doubleday.

Olson, J., & Moran, T. P. (1996). Mapping the method muddle: Guidance for using methods in user interface design. In M. Rudisill, C. Lewis, P. Polson, & T. McKay (Eds.), *Human-computer interface design: Success stories, emerging methods and real-world context* (pp. 269–302). New York: Morgan Kaufman.

Payne, S., & Green, T. R. G. (1989). Task-action grammar: The model for and its development. In D. Diaper (Ed.), *Task analysis for human-computer interaction* (•••). Chichester, UK: Ellis Horwood.

Ramey, J., Rowberg, A. H., & Robinson, C. (1996). Adaptation of an ethnographic method for investigation of the task domain in diagnostic radiology. In D. Wixon & J. Ramey (Eds.), *Field methods casebook for software design* (pp. 1–15). New York: Wiley.

Redish, J. C., & James, J. (1996). Going to the users: How to set up, conduct, and use a user and task analysis for (re)designing a major computer system. *Proceedings of the Fifth Annual Conference UPA '96*, Dallas, TX, •••.

Redish, J. C., & Wixon, D. (2003). Task analysis. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 922–940). Mahwah, NJ: Lawrence Erlbaum Associates.

Siegel, D., & Dray, S. (2005). Making the business case for international user centered design. In R. Bias & D. Mayhew (Eds.), *Cost-justifying usability: An update for the Internet age* (•••). San Francisco, CA: Morgan Kaufmann.

Simpson, K. T. (1998). The UI war room and design prism: A user interface design approach from multiple perspectives. In L. Wood (Ed.), *User interface design: Bridging the gap from user requirements to design* (pp. 245–274). Boca Raton, FL: CRC Press.

Uren, E. (1997). Annotated bibliography on internationalization and localization. *Journal of Computer Documentation* (*SIGDOC*)*, 21*(4), 26–33.

Whiteside, J., Bennett, J., & Holtzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.), *The handbook of human computer interaction* (pp. 791–817). New York: North Holland.

Wixon, D. (1995). Qualitative research methods in design and development. *Interactions, 2*, 19–24.

Wixon, D., Holtzblatt, K., & Knox. S. (1990). Contextual design: An emergent view of system design. *Proceedings of CHI'97,* Location, 329–336.

Wixon, D., & Comstock, E. (1995). Evolution of usability at Digital Equipment Corporation. In M. Wicklund (Ed.), *Usability in practice: How companies develop user friendly products* (pp. 147–191). New York: Academic Press.

Wixon, D., & Jones, S. (1996). Usability for fun and profit: A case study of the design of DEC Rally Version 2. In M. Rudisill, C. Lewis, P. Polson, & T. McKay (Eds.), *Human-computer interface design: Success stories, emerging methods and real-world context* (pp. 3–36). New York: Morgan Kaufman.

Wixon, D., & Ramey, J. (1996). Field oriented design techniques—case studies and organizing dimension. *SIGCHI Bulletin, 28*(3), Retrieved date, from http://www.acm.org/sigs/sigchi/bulletin/1996.3/wixon.html

Wood, L. (1996). The ethnographic interview in user-centered work/task analysis. In D. Wixon & J. Ramey (Eds.), *Field methods casebook for software design* (pp. 35–56). New York: Wiley.

·49·

CONTEXTUAL DESIGN

Karen Holtzblatt
InContext Enterprises, Inc.

Introduction **950**
Rapid Contextual Design Overview **950**
Setting Project Scope: What Makes a Good Rapid CD Project? 951
The Steps of Rapid Contextual Design 952
The Contextual Interview—Getting the Right Data 952
Choosing customers depends on project scope 952
Using content experts 953
Interpretation Sessions and Work Modeling—Creating a Shared Understanding 953
Working as a distributed team 954
Consolidation—Creating One Picture of the Customer 954
How can so little data characterize a whole market? 955
Personas give voice to the consolidated data 956
You cannot invent from customer data 958
Defining Your Rapid CD Process—The Half-Way Point 958
Storyboarding—Working Out the Details 958
How is storyboarding different from building scenarios? 959
The User Environment Design option 959
User Interface Design and Mockup 960
Why build initial paper prototypes on paper—paper versus online testing 961
How are mock-up interviews different from demos, rapid prototyping, or usability tests? 961
Integrating Contextual Techniques Into Standard Processes **961**
General Corporate Methodologies 962
Rational Unified Process 962
Agile or Extreme Programming 962
Conclusion **963**
References **963**

INTRODUCTION

Contextual Design (CD; for an introduction to all the steps of CD, see Beyer & Holtzblatt, 1998) is a customer-centered design process that takes a cross-functional team from collecting data about users in the field, through interpretation and consolidation of that data, to the design of product concepts and a tested product structure. Over the last 18 years, the industry has moved from using an engineering-driven requirements and design process to a more user-centered process. Many now recognize that the best way to define the right product, business process, website, or other system is to involve users and user data at every step. CD has been used in companies and taught in universities all over the world, along with other user-centered design processes. Today, enlightened product companies and IT organizations know that successful projects and user adoption come from using user-centered design processes.

But along with a commitment to working with users has come a demand to reduce the time needed for up-front design. Teams want to include user data without adding significant time to the process. Organizations have existing methodologies and want user-centered design techniques to be included within these existing practices. Opposition to bringing users into the center of the design process often centers on the lack of time in the schedule.

CD is a framework for structuring front-end design to ensure that user data drives the overall system definition as well as the user interface. As such, a CD project may range from weeks to months depending on the scope and goals of the project. In our experience, complaints about time are less about overall clock time and more about how user data fits into the existing habits, processes, job descriptions, and schedules of a company. To help companies see how to use the steps of CD rapidly we have put together a structure for how to use the techniques in one to seven weeks. Rapid CD tries to answer the questions so often asked about the full CD process:

- Do I have to do all those steps? What can I skip? When can I skip it?
- How does CD fit with my existing design process? Can I use CD techniques to get customer data and then use the steps I am accustomed to using?
- I have only two people on the team. Can I still do it?
- What can I do in a few weeks?

Any requirements gathering and design process takes time. For the CD process, the real speed (clock time) to project completion depends on the following variables:

- The number of customer visits that you choose to perform
- The number of people who can work simultaneously on the project or the helpers you can get at key points
- The dedication of the people assigned to the project and whether they can work full-time
- The size of the problem—the more complex the business process, the more complex the product and therefore the longer it will take to define or redesign
- The number of stakeholders that have to be satisfied, coordinated with, and communicated to and so the more buy in you need and the longer it will take

With a small enough project and focused, trained, and dedicated resources, teams can run projects that infuse customer data into the design in a few weeks. This chapter provides an overview of the steps of Rapid CD and key issues in its use.

RAPID CONTEXTUAL DESIGN OVERVIEW

We have identified several viable Rapid CD design processes to guide teams in choosing a process right for their projects. Table 49.1 provides an overview of the steps used in each of these processes.

- **Lightning Fast.** Characterize your user population and vision a solution in one to four weeks.

TABLE 49.1. Comparison of Rapid CD Processes

Rapid CD Process	Contextual Interviews with Interpretation	Sequence Model with Consolidation	Affinity Diagrams	Wall Walk & Visioning	Storyboarding	Paper Mock-up Interviews with Interpretation
Lightning Fast 1–4 weeks	4–12 users					
Lightning Fast + 4–8 weeks	6–12 users					4–9 users
Focused 6–10 weeks	8–12 users					6–12 users

- **Lightning Fast+.** Characterize your user population, vision a solution, mock it up in paper, and test it in four to eight weeks.
- **Focused Rapid CD.** Characterize your user population, collect and consolidate sequences for a task analysis, vision a solution, work out the details by storyboarding, mock it up in paper, and test it in 6 to 10 weeks.

Rapid CD processes assume a two-person team with occasional helpers at key points. No matter the time or resource constraints, there is always time to get user data into the process—if the organization is committed to it. These Rapid CD project types provide some guidelines for structuring quick processes for any project. Successful projects always need to have strong project management and must be run on day-by-day schedules (for detailed descriptions of each step, schedules and guidelines for project management, see Holtzblatt, Wendell, & Wood, 2005; see Table 49.2 for a sample schedule for a one-week project).

Setting Project Scope: What Makes a Good Rapid CD Project?

Rapid CD techniques can be used on any project to gather customer data. However, some projects are especially well suited for Rapid CD. The basic rule of thumb is to have a small, tight project scope targeting a few clearly defined job roles. Following are some example projects that work well:

- **Usability fixes, low-hanging fruit, quick fixes.** Sometimes a project or system is just looking for the top fixes to take a significant step to improve the product or system. You can observe a cross section of key users interacting with the existing product, system, or website. From this data, top recommendations easily can be generated.
- **Market or population characterization for new system concepts.** The goal of this kind of project is to understand the users and generate ideas for new system directions. If your new market or business process has no more than four job roles in the target population or process, you can use Rapid CD to characterize the needs of the population.
- **Website evaluation and redesign.** You can use Rapid CD to redesign your current website to provide your users better access to information and services if your target user population covers one to three major job roles. Rapid CD can evaluate how people currently are using your website and other related sites and help set new direction.
- **Next-generation system.** If you want to redesign an existing product (not a product suite) and your product targets one or two major job roles, Rapid CD will help you to identify new function, to grow the product, and to improve the user experience.
- **Supporting a coherent task.** If you want to build a system to support a coherent task that may be accomplished by one to four collaborating job roles, Rapid CD can help. This task may be part of an existing system, a larger business process, a task within a larger ERP environment, a portal, a product, or a website.
- **Persona development.** If you want to develop personas for your website, product, or process and you have one to four target job types, Rapid CD can provide you with the data you need to find your user types and write rich personae.

Rapid CD can work for these projects because the scope is tight. This makes Rapid CD a good companion for Agile development techniques where the scope is already tight.

The wider the scope is, the longer the process is, and the more resources are required. Therefore, a wider scope like a large business process or suite of products is not appropriate for Rapid CD techniques when addressed together. However, when these projects are broken into tasks or individual products and addressed successively, you can use Rapid CD techniques for successful redesign. If you use the same project team for the whole succession of projects, you will end up with a coherent system. If you try to work multiple teams in parallel, you may inappropriately break up the work or fragment the design.

TABLE 49.2. Sample Lightning Fast schedule for Six Customers, Requiring Five to Six days to Gather, Interpret, and Organize the Data and Bainstorm Solutions. Preparation for Visits and Presentation of Findings is Not Included

Monday	Tuesday	Wednesday	Thursday	Friday
AM Two parallel customer interviews. Return immediately for interpretation.	**AM** Same as Monday.	**AM** Same as Monday.	**All Day** Build the affinity of approximately 300–400 affinity notes. Include at least 2–4 more people to help. With no helpers, 2 people will need Friday to finish. This will extend the schedule by 1 day.	**All Day** Finish any clean-up of the affinity by 11:00 A.M. Walk the affinity to generate design ideas. Brainstorm recommendations as a group. Capture recommendations online during the session. Formal preparation of recommendations is not included.
PM Two back-to-back interpretation sessions. Bring in a third person for perspective and buy-in. Finish both interpretations.	**PM** Same as Monday. Discuss finding and reset focus to ensure the quality of the data.	**PM** Same as Monday. Finish all interpretations by end of day.		

The Steps of Rapid Contextual Design

The Contextual Interview—Getting the Right Data

In order to design a product that meets customers' real needs, designers must understand the customers and their practices.[1] Yet designers are not usually familiar with or experienced in the user activities they are supporting. If they operate from their gut feelings, they rely on their own experiences as users. Generally, designers are more tolerant of technology than average users, so they are not representative of end users.

On the other hand, requirements gathering is not simply a matter of gathering requirements by asking people what they need in a system the way pebbles can be gathered from the beach. A product is always part of a larger practice. It is used in the context of other tools and manual processes. Product design is fundamentally about the redesign of work or life practice, given technological possibility. In addition, practice cannot be designed well if it is not understood in detail.

You cannot simply ask people for design requirements, in part because they do not understand what technology is capable of, but more because they are not aware of what they really do. Because the everyday things people do become habitual and unconscious, people are usually unable to articulate their practices. People are conscious of general directions, such as identifying critical problems, and they can say what makes them angry at the systems. However, they cannot provide day-to-day details about what they are doing. This low-level detail of everyday practice grounds designers in what is needed before they invent how technology might augment the process.

The challenge of getting this design data is finding a technique to get at data that is unconscious and tacit. The first step of CD is Contextual Inquiry, our field data gathering technique that allows designers to go out into the field and talk with people about their work or lives while they are observing them. If designers watch people while they work, the people do not have to articulate their practices. If they do blow-by-blow retrospective accounts of things that happened in the recent past, people can stick with the details of cases using artifacts and reenactments to remind them of what happened. Field data overcomes the difficulties of discovering tacit information.

In CD, the cross-functional design team conducts one-on-one field interviews with customers in their workplaces (or life spaces) focusing on the aspects of the practice that matter for the project scope. The Contextual Interview lasts about two hours and is based on four principles that guide how to run the interview:

- **Context—**While people are working, gather data in the workplace and focus on the activities they are doing.
- **Partnership—**Collaborate with customers to understand their work; let them lead the interview by doing their work. Do not come with planned questions.
- **Interpretation—**Determine the meaning of the customer's words and actions together by sharing your interpretations and letting them tune your meaning. When immersed in their real lives and real works, people will not let you misconstrue their lives.
- **Focus—**Steer the conversation to meaningful topics by paying attention to what falls within project scope and ignoring things that are outside of it. Let users know the focus so they can steer, too.

The Contextual Interview starts like a conventional interview, but after a brief overview of the practice, it transitions to ongoing observation and discussion with the user about that part of the practice that is relevant to the design focus. The interviewer watches the customer for overt actions, verbal clues, and body language. By sharing surprises and understandings with users in the moment, users and designers can enter into a conversation about what is happening, why, and the implications for any supporting system. As much as possible, the interviewer keeps the customer grounded in current activity, but can also use artifacts to trigger memories of recent activities.

The fundamental intent of the Contextual Interview is to help designers get design data: low level, detailed data about the structure of the practice and the use of technology within that practice. At InContext, we believe that the Contextual Interview, which is based on observing people in the context of their practices while they do their normal activities, is the best way to get this necessary design data. This data is critical—whatever else we trade off in Rapid CD, we never trade off the field interviews.

Choosing customers depends on project scope. The number of people that should be interviewed is directly related to the project scope. The wider the scope, the more people need to be interviewed to cover that scope. Our rule of thumb for a small scope, such as top 10 problems, usability improvement, next product release, or checking a planned design, is 6 to 10 customers from three to five businesses covering one or two roles. The more roles you need to cover and the more contexts (type of business, characteristic of person, geographic location), the more people you need to interview.

When analyzing the project, first identify the job titles or roles targeted and the people that support the activities to be studied. A Rapid CD project can handle one to four roles, which means keeping the scope of the practice tight. Remember, the work or life practice users engage in is what counts—if one person is called a system administrator and the other a database administrator, but they do the same work as it relates to the project, it does not matter that their job titles are different. We interview a minimum of three people per role; four is better.

If the product is to support people across different contexts, for example industries, what matters is whether the contexts imply a different practice. In real estate, for example, the work is structured differently within the industry: a group of small, distributed agencies, a large corporate real estate company, and

[1]Some products, systems, and websites support the way people work, keep businesses running, or help users find needed information. Other products, systems, and websites address games, other entertainment, or consumer information to support life decisions. To gather data for these consumer products we have to look at people's life practice. To simplify language, this chapter will use *practice* to mean both work and life practice.

an in-house real estate representative. In each situation, the communication, sharing, and work management is likely to differ, creating three different practice patterns. To cover real estate, you need to collect data in all three contexts. We try to interview at least two businesses or independent set of people per practice pattern.

If, for your industries, there are no changes to practice patterns, then simply touch multiple industries without worrying about overlap. In this case, wide diversity is best.

The goal in selecting users to interview is to get enough repetition in the practice so that each role and contextual variable has three or four interviews that represent it, remembering that any one person may represent several of the contextual variables. As long as you have overlap, you will be able to find the common structure and key variations in the practice. Remember, paper prototype interviews will expand the number of contexts and roles represented in the whole project.

Using content experts. A user advocate is not the user. The core to your success is to get to the real users of your proposed system. Some organizations resist letting teams go to the user because they think they have in-house expertise. Some companies hire business analysts who used to be users in the hope that they will provide the requirements knowledge needed. In addition, some methods, such as XP, advocate that a customer representative is all you need to iterate the design.

But a representative user is not the same as a set of current actual users. A representative user never is. No one person can embody all the users of a real system. Although interviewing several users reveals a common practice, no one user ever represents that practice in its entirety. Furthermore, the more the user advocate becomes part of the engineering organization, the less useful they are as a user surrogate. They learn too much about the technology, and they become invested in the team's thinking.

More importantly, user advocates cannot articulate their practices any more than users can. So involve user advocates in your process, put them on the team, and let them tell you their entering assumptions about the practices and problems to help set focus for the interviews. However, do not use them as substitutes for real user data.

User advocates do not speed the process—they just create another in-house person with whom to argue. Only data collected through an agreed upon external process will get respect for the findings across the organization. You speed the process by collecting the right data and generating real findings through an agreed upon method.

Interpretation Sessions and Work Modeling—Creating a Shared Understanding

Contextual Interviews produce large amounts of customer data, all of which must be shared among the core design team and with the larger, cross-functional team of user interface designers, engineers, documentation people, internal business users, and marketers. Traditional methods of sharing through presentations, in reports, or by e-mail do not allow people to truly process the information or bring their perspectives into a shared understanding. CD overcomes this by involving the team in interactive sessions to review, analyze, and manipulate the customer data. For Rapid CD your two-person core team should be cross functional: for example, a user researcher and an interaction designer; a product manager and a developer; or an analyst and an interaction designer.

The data will be richer and the process of buy in will be enhanced if other stakeholders who will be building, marketing, or using the system participate in the interpretation session. This is a two-hour commitment, a reasonable request for help in most organizations. We therefore recommend a three-person team for interpretation sessions in Rapid CD.

While we audio record interviews for backup, we do not transcribe the recordings or do videotape analysis. Both transcription and video analysis take too long, and video analysis limits the perspective to one person. Instead, within 48 hours of each customer interview, the team gathers in the design room, where the interviewer tells the story of the interview from handwritten notes and memory.

Team members ask questions about the interview, drawing out the details of this retrospective account. One person is the recorder typing notes online (or into CDTools[2]). The other team member who did the interview retells the story being captured. The helper listens and points out key issues to be captured. Each team member brings a different perspective to the data, while open discussion enables the team to arrive at a shared understanding. Participants ask questions, triggering the interviewer's memory and eliciting more data than would be available from a designer working alone. When the discussion sparks design ideas, they are captured in the notes.

The notes are displayed on a monitor or computer project so that everyone can see them. These notes capture the key practice issues, cultural observations, breakdowns, successes, task patterns, design ideas, and any other interpretation or issue that has relevance to the project scope. Later these notes are transferred to Post-its and used to build the affinity diagram.

For some Rapid CD projects, all you need are these notes to characterize the problems and issues of the customer population. But, if you want a task analysis, plan to do storyboarding, or need to map a work process, we recommend you also capture the CD sequence model. In Focused Rapid CD, the team captures the sequence model showing each step required to perform a task in order.

Rapid CD does not use any of the other four work models from the larger CD process. Capturing and consolidating work models characterizes the population much more completely but takes time to build. Rapid CD trades off this richness for time. Because the scope of a Rapid CD project is small and tight, it is already reasonably tactical making the richness of additional data nice to have but not necessary.

The interpretation session is a structured way to ensure that all relevant information from the interview is captured for use in

[2]CDTools assists a Contextual Design team in gathering and organizing CD data. It provides features to track users interviewed, capture interpretation notes, capture an affinity diagram, and publish the organized affinity to the Web. [AQ17]

the design process and shared with key team members. It is the context for both understanding and seeing the structure of the data and starting a real design conversation about how to address the users' needs with technology.

Working as a distributed team. More and more companies are expecting their teams to be distributed. Distributed teams have the advantage that data can be collected from different locations with less travel, which saves time and cost. However, any distributed team has to be more organized, relies on the latest collaborative technology, and needs to work harder to develop a shared understanding.

The interpretation sessions of Rapid CD are easy to do in a distributed fashion. If you have collaboration or virtual meeting software, you can use that to run a distributed meeting. Simply display CDTools or your word processor in a collaboration space so all can see the notes as they are typed. Get everyone on the phone and start the session as usual. As the interviewer tells the story, the note taker captures the issues and others can call out design ideas and ask questions to get at the details of what happened.

If you are capturing sequence models, you may want to consider capturing those online in a word processor, spreadsheet, or drawing tool, switching the shared display from the interpretation notes to the sequence model as needed. Periodically stopping to check the sequence models will add to your interpretation session time, but is a critical step to be sure that the data is being captured correctly.

The key to success is keeping everyone engaged. Be sure that everyone has a role and is actively participating. If someone has been quiet for a period, check in with him or her. Consider having both sides of the phone responsible for some type of data capture to ensure involvement.

Finally, do not start distributed. Do your first interviews on the road together or in one location. Work out your process and your initial understanding of the users' activities, issues, and practices together. This becomes the bedrock of your shared understanding. After that, the distributed session works very well.

Consolidation—Creating One Picture of the Customer

Consolidating the notes and the sequence models creates a coherent representation of a market's or a user population's needs and activities. Then the design can address a whole population, not just one individual. The most fundamental goal of CD is to get the team to design from data instead of from the *I*. If you walk in the hall and listen to designers talking, you will hear comments such as "I like this feature" or "The interface will work best this way." It is rare to hear, "Our user data says that the work is structured like this, so we need this function." It is natural for people who design to make a system hang together in a way that makes sense to them. But they are not the users, and increasingly, they are not in any way doing the activities of the people for whom they are designing. Getting designers, marketers, and business analysts out to the field and into interpretation sessions moves them away from design from personal preference. However, we do not want them to become attached to their users to the exclusion of the rest.

Product and system design must address a whole market or user population. It must take into consideration the issues of the population as a whole, the structure of work, and the variations natural to that work. The core intent of data consolidation is to find the issues and the activity structure and create a coherent way to see it and talk about it.

In Rapid CD, we consolidate the notes from the interpretation session into the affinity diagram and the sequences into consolidated sequence models.

The affinity diagram (see Fig. 49.1) brings issues and insights across all customers together into a wall-sized hierarchical diagram. The team prints the notes captured in the interpretation session onto labels that are sticky notes. The team then organizes all this data into groups, finding common underlying themes that cross the customer population. The process ex-

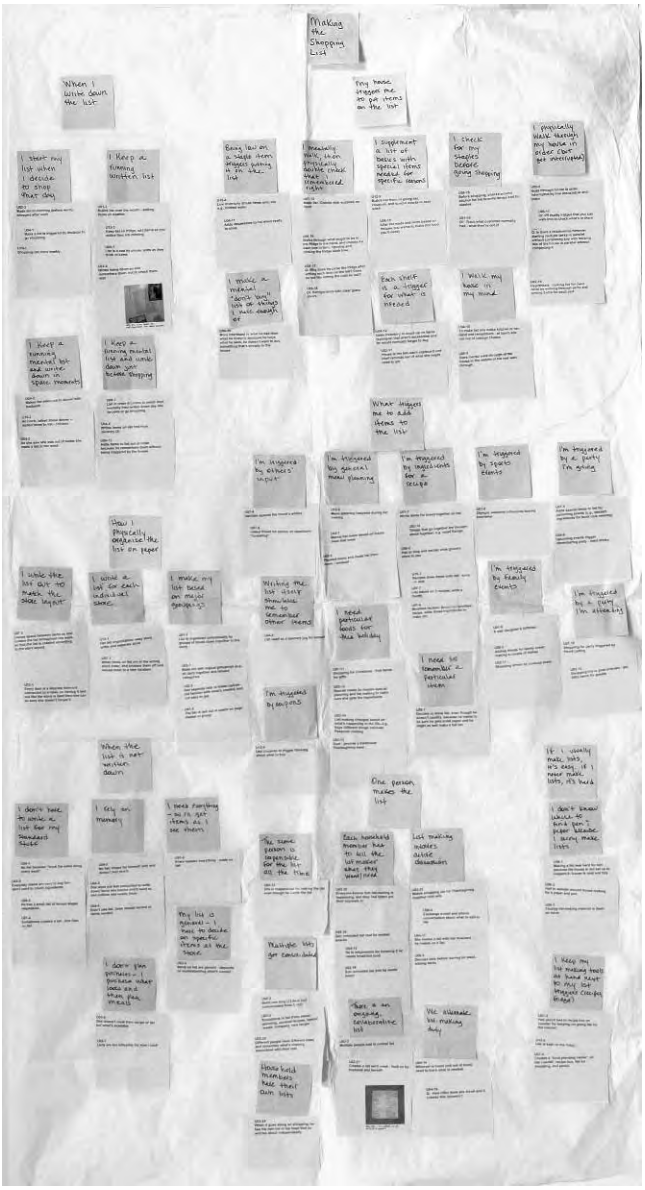


FIGURE 49.1. The affinity diagram

poses and makes concrete common issues, distinctions, practice patterns, and needs without losing individual variation. Walking the affinity diagram allows designers to respond with design ideas that extend and transform the work. They write these on Post-its and stick them right to the data that stimulated the idea. This encourages a culture of design from data over a focus on cool ideas generated from the *I*.

The affinity is built from the bottom up, which allows the individual notes that come together to suggest groupings instead of trying to force them into predefined categories. Groups are labeled using the voice of the customer—saying what they do and how they think. In Rapid CD, a team of two can build an affinity diagram in one to three days. This is the place that we recommend using helpers—again, a one-day commitment can speed the process and expose stakeholders to the raw data building buy in to the process and the data.

The consolidated sequence model (see Fig. 49.2) shows the detailed activity structure that the system will support or replace. This is similar to traditional task analysis, showing each step, triggers for the steps, different strategies for achieving each intent, and breakdowns in the ongoing work. Work practice redesign is ultimately about redesigning the steps in the sequences. Whether the redesign eliminates or changes the steps or eliminates the whole sequence, knowing the steps and intents keeps the team honest. More often than not, technology introduces new problems into the practice by failing to consider the fundamental intents that people are trying to achieve. Redesign will better support the practice if it accounts for each intent, trigger, and step. This does not mean leaving the steps of the activity the same; it means that the team has seen the current activity and has completely considered what will happen to it in the new redesigned practice. The consolidated sequence is critical as a guide for storyboarding or for streamlining any business processes.

Shopping in the Store
-Activity 2. -Stop and Buy

1) Trigger: Stop if:	
a. Passing a block of stuff needed, or staple, or treat (may or may not be on list)	▪ Get ingredients for recipes ▪ Please co-shopper (especially kids) by buying them a treat
b. Reached a home base where you re willing to browse, park cart and wander	▪ **Many shoppers scan items, not signs; signs are invisible to them even if they advertise specials they care about** ▪ Allow store to suggest purchases after I've decided I'm in an area of the store where I'll allow that ▪ Save pushing the cart around by parking it and using it as a home base while I wander from item to item
c. You want to check on prices for future reference (may or may not buy)	▪ Save money on staple item
d. You notice sale item and want to decide if it meets need	▪ Save money on frequently used item, even if not on list
2) Use block of stuff to suggest possible meals	▪ Plan the week s meals taking into account what is in the store and what looks good
a. Discuss menus with other shoppers	▪ Share the meal planning ▪ Plan who will cook what
3) Decide whether to get stuff from that block, considering:	▪ Choose a food that supports my values
a. What exact type and brand to buy, what family will like, size, weight and amount to buy, cost, quality, ripeness, freshness, good packaging, whether I have coupons, and ingredients	▪ Compare and learn quality of new items ▪ Minimize sugar/fat consumption
4) Put items in basket	
5) Or, discover store doesn t carry desired item	
6) When done with the block of stuff, check to see where you are in the aisle	▪ Reorient yourself after buying from a block of stuff
a. See something, remember need — Triggers other items you need	▪ If blocks on opposite side of aisle are staggered, may have to backtrack
b. Backtrack if missed something on the other side of aisle, or	
c. If you re reminded of something you forgot	▪ Get items quickly
1) Go get the item (Beeline shopping strategy)	
2) Go back to the cart and make sure everyone s OK	
3) Put item in basket	

FIGURE 49.2. The consolidated sequence model.

Consolidate sequences by collecting together sequences with similar overall intents. Then, group the steps into chunks that represent activities and consolidate across like activities. Create abstract steps that represent the detailed individual steps in each actual sequence model. Consolidate only the tasks that the system will support or that you need to understand in detail.

How can so little data characterize a whole market? Consolidated data is built from 8 to 30 field interviews and can characterize markets of millions of people. Consolidated data from 8 to 12 people can identify a large percent of the key issues that will eventually be identified in the population. For example, even on large projects we routinely consolidate after 10 to 15 interviews and then grow the models and affinity from there as more data is collected. An early affinity represents the key areas and distinctions that will grow in detail as more data is added, and reveals holes in the data to guide further data collection. Similarly, additional data adds depth and detail to the sequence model, but the basic structure, central to the project focus, is identifiable early. Therefore, a Rapid CD project with 8 to 12 key users will reveal the vast majority of the issues that you need to consider for your project.

Early in usability testing, people were not comfortable with test results from small numbers of users. Whatever the formal arguments, years later we have learned empirically that small numbers are enough. CD gets its power from designing from an understanding of practice structure without losing variation.

Consolidation helps people see and find the structure in the practice, and this drives successful design. Here is a way to think about it.

We are all different, but we are alike. Everybody looks different; humans have great variation. People are of different ethnic groups, cultures, and child rearing practices. Everyone chooses different clothes, hobbies, careers, and life styles. At one level, we are all unique. However, at the same time we are alike, each having one head and body with two arms and two legs. For example, clothing manufacturers can make off the rack clothing; making a few adjustments for different body sizes is enough. Structurally, the variation between people is small and the structure of our bodies is common.

There are only so many ways to do anything. Any product and system design is really a very narrow focus on the human experience. Within one kind of work or activity, only so much variation is possible. The roles we play, the intents and goals we have, and the way we do things is similar. Variation, once you start looking for structural elements is small. We have found, for example, that there are only two to four strategies for any primary task in a practice. If those are the key strategies for that practice, the question is not which we support, but how we support them all.

Designs that target a certain domain of activity are created for a certain set of people doing a fixed set of tasks, situated in the larger culture, using the same set of tools, trying to achieve the same kind of goals. Under these constrained conditions, practices from person to person will have similar patterns. After you have collected data from three to six people doing the same thing, the activity patterns and issues start overlapping.

Personas give voice to the consolidated data. When your cross-functional team has worked together to gather, interpret, and consolidate customer data, you have developed a shared understanding of the users and their issues. You have not just built the data—you have experienced it in a way that others have not. Therefore, you may find the consolidated models and affinity to be evocative representations of the customer data you experienced, but others do not share that experience and cannot assimilate these models as easily.

Most teams need to communicate their understandings of user needs and their design plans to stakeholders—management, customer organizations, product groups, and so forth. These stakeholders, with no background in CD and no experience with the data, do not embody the memories of the real users from the field interviews. When they walk the data they can see the issues, needs, and breakdowns, but the power of knowing the users and their issues personally is not as poignant or personal.

Personas can help bring users alive and focus the stakeholders on the relevant issues when they are built from rich contextual data. Popularized by Cooper (1997), a persona describes typical users of the proposed system as though they were real people. Their use is becoming more widespread, though with mixed success. According to Manning (2003), "A persona that's not backed by rich contextual data isn't valid, which accounts for much of the mixed success." However, when backed by rich contextual data, they can help developers and designers not involved in the data collection focus on the needs and characteristics of their users (Manning, 2003). Anyone, trained in CD or not, can read this vignette and gain a sense of the typical user they are trying to support.

We build personas from the data collected in Rapid CD to help focus on the characters we are visioning about in the next step, to help stakeholders segment their market according to practice instead of typical demographics, to clarify branding and prioritization, and to bring the users and their needs to life for developers.

To build a persona, the team looks for core practice roles among the users, each characterizing a different way of doing the work. Expect anywhere from two to six of these in a Rapid CD project. Then for each core practice role, find the base user: the user that most exemplifies this role. Look at others who also manifest this role and borrow other relevant tasks, values, and life story elements to create your archetypical role. This collection of exemplary facts then becomes the basis for your story of the named persona. With the rich user data in the affinity already grouped into issues, it is easy to harvest the data for the key elements that differentiate the different personae. Write a paragraph about of the persona's project-related life; get a representative picture; list the typical tasks, roles, goals, and other practice characteristics relevant to your project focus, and you have built a persona based on reliable customer data that you know represents your user population.

Visioning a New Work Practice

Visioning is about invention. However, design of technology is first design of the story showing how manual practices, human interactions, other tools come together with your product or system to better support the whole practice. Visioning is the CD technique to help teams tell that story. Visioning is a vehicle to identify needed function in the context of the larger practice. Visioning ensures that teams put off lower-level decisions about implementation, platform, and user interface until they have a clear picture of how their solution will fit into the whole of the practice. Teams commonly focus too much on low-level details instead of the full sociotechnical system. This is one cause of breaking the way users perform activities and of failing to create something the market wants. Therefore, the primary intent of visioning is to redesign the practice, not to design a user interface. Because a visioning session is a group activity, it fosters a shared understanding among team members and helps them use their different points of view to push creativity.

In Rapid CD, the core team visions a solution, but this is another time for bringing in stakeholders and helpers. A good visioning session has from four to six participants. If helpers have been involved in the interpretation session and affinity building they may want to participate in the visioning session. Again, this is a one- or two-day commitment, which allows a wider team to participate in creating the solution direction.

The first step to a visioning session is to walk the data model by model, immersing the team in customer data so their inventions will be grounded in the users' work. During the walk, team members compare ideas and begin to get a shared idea of how to respond to the data. Our rule is that no one gets to vision unless he or she has participated in walking the data. Without this rule, the process is no longer data driven; anyone can walk in and offer their pet design ideas. Simply walking the customer data naturally selects and tailors preexisting ideas to fit the needs of the population. Since we will evaluate the visions based in part on fit to the data, knowing the data is important.

During the visioning session, the team will pick a starting point and build a story of the new practice. One person is assigned to be the pen, or the person who draws the story on a flip chart, fitting ideas called out by the team members into the story as it unfolds. The story describes the new practice, showing people, roles, systems, and anything else the vision requires. The team does not worry about practicality at this point; all ideas are included. Creating several visions allows the team to consider alternative solutions.

After a set of visions is created, the team evaluates each vision in turn, listing both the positive and negative points of the vision from the point of view of customer value, engineering effort, technical possibility, and corporate value. The negative points are not thrown out but used to stimulate creative design ideas to overcome objections. When complete, the best parts of each vision and the solutions to objections are brought together into one, synthesized practice redesign solution (see Fig. 49.3).

You cannot invent from customer data. Every team we work with raises this claim. Customer data tell you what is, not what could be. How can you see the future by looking at the past? Here is what we say.

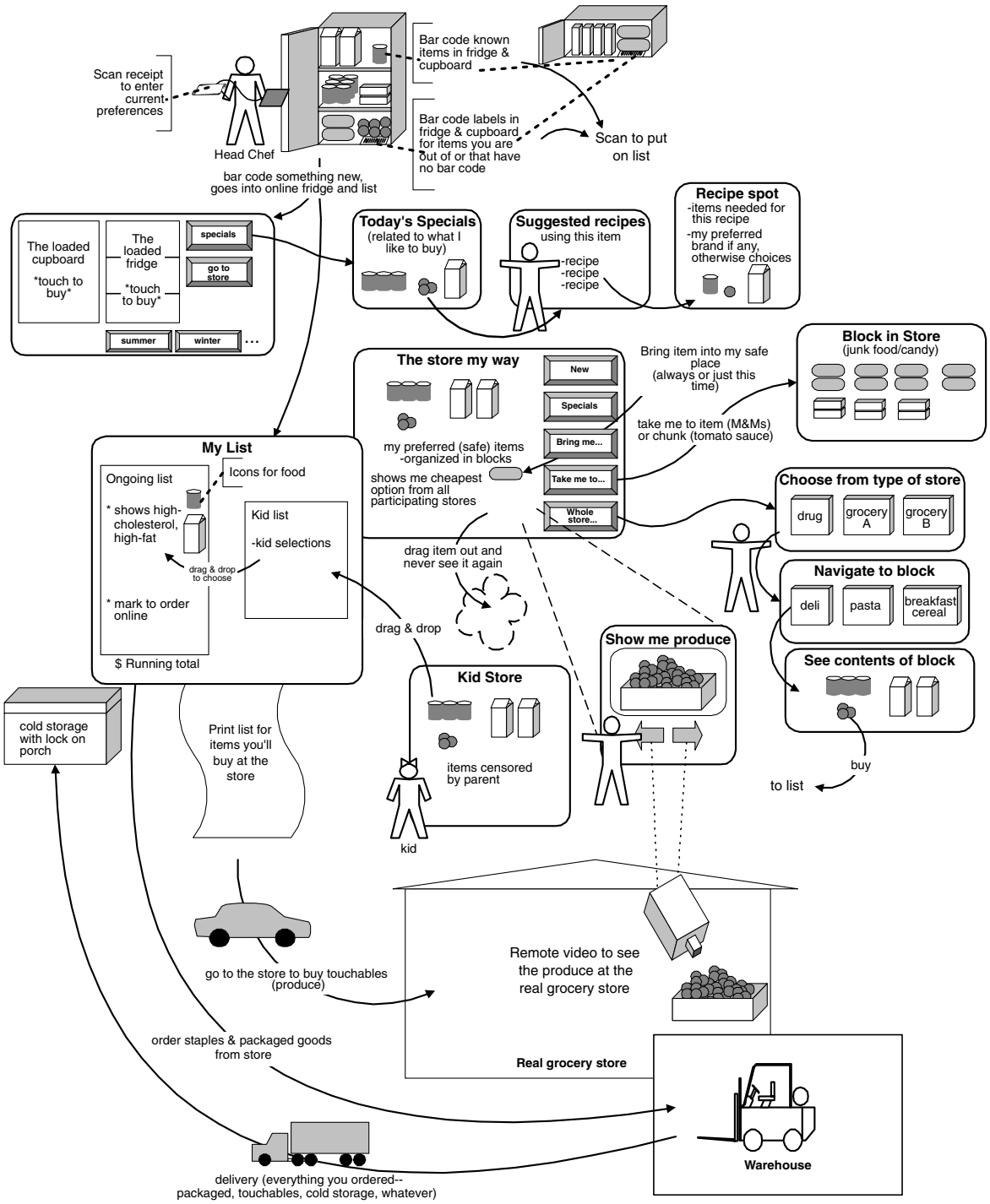


FIGURE 49.3. A complete, synthesized vision.

Every invention supports a real need, otherwise why would anyone want it? Invention is a response to some life or work practice by a designer or technologist who, seeing a need *and* knowing the technology, imagines a new possibility. Edison did not invent the idea of light; he saw candles and gas and invented light bulbs. Bricklin did not invent spreadsheets; he saw how paper spreadsheets are used and knew what technology could do. The developers of WordPerfect worked in the basement below a secretarial pool. Nobody invents entirely new things that fulfill no need and contribute in no way to human practice; they invent new ways to fill existing needs and overcome existing limitations. As people incorporate these products and new ways of working into their lives, they reinvent it by adopting and adapting the new way of working. If designers are out there seeing

people living their lives with, without, and in spite of technology, they can see future directions for technology.

A vision is only as good as the team's combined skill. Customer data is the context that stimulates the direction of invention. However, no invention is without understanding the materials of invention: technology, design, and practice patterns. The visioning team needs to include people who understand the possibilities and constraints of the technology. If the team is supposed to design Web pages and none of them has ever designed a Web page, they will not be able to use Web technology to design. When the people who always designed mainframes were told to design windowing interfaces, they replicated the mainframe interface in the windowing interface. This is why we recommend that design teams include people with diverse backgrounds representing all the materials of design and the different functions of the organization. Only then will an innovative design, right for the business to ship, emerge.

We do not ask customers what to make; we understand what they do. Customers who are not aware of the details of their practices, who do not know the latest technologies, and who do not know what your business is capable of cannot tell you what to invent. We do not ask them. Instead, we understand what they are doing and capture it systematically. We immerse a design team—who does understand technology, the practice, and the business—in that data and let them vision. However, we are not done. The vision has to be right for people, so we take it out and test it. We let people test drive the future in our paper mock ups and let their tacit knowledge of their lives shape and direct our vision.

Defining Your Rapid CD Process—The Half-Way Point

The vision, produced by a cross-functional team immersed in customer data, is the jumping off point for concurrent design:

- **Interaction design:** The vision guides the detailed design to produce the overall user experience plan and final interaction design.
- **Engineering:** The vision contains implementation assumptions and challenges that must be looked at for viability before the company can commit to the vision. The data provides the basis for as-is use cases that can be used to characterize the current practice.
- **Marketing:** The vision is the story of the new practice—the basis for communication to customers. Sales and marketing are always based on storytelling: how the new product will benefit the customer. Sharing the story in user groups, focus groups, and individual conversations is a good way to gauge sales point (the excitement that the story will generate).
- **Business planning:** The vision can be used to drive marketing surveys and investigations to flesh out the business case.
- **Business (or enterprise) process design:** The data itself reveals root causes to problems and drives which processes to study more and measure. It also provides a representation of the as-is process to drive process mapping and as-is use case development. The vision represents the new sociotechnical system that could be put in place to streamline the process. Rather than starting by improving the processes and then designing the system to support it, the vision redesigns the process and the technology as one integrated whole. This speeds the design process and ensures that the systems and process support each other in creating a new practice that people are willing to adopt.
- **Testing:** The consolidated models drive test case development.
- **Documentation:** The vision communicates what the product or system is so they can start the introduction to the user's manual.

Because the vision is the center of the design, revealing direction for a system or product based on real user data, some companies may wish to transition to their existing processes after this step. Lightning Fast CD gets customer data into the product or system-concepting process and relies on the organization's existing process to define and test the actual system.

Lightning Fast CD is also a good starting point for an organization—in only a few weeks, you can gather and organize data to reveal problems, issues, root causes, and opportunities. The vision step helps teams organize their responses to this data to create a direction for their company or project. This direction can be tactical or identify new product direction depending on the project scope. As such, Lightning Fast CD infuses user data into a design process simply and easily without challenging other existing practices that might be more well defined, already successful, or simply rooted in history and so hard to change.

Lightning Fast CD also supports some of the new trends in design. For Agile techniques, the vision implies core function and implicit prioritization, which can be broken into stories for an Agile planning session. If a company simply wants to develop personas, the consolidated data provides the information needed to flesh out the characters your organization needs to support. If you add visioning to this, you have stories of how these characters might operate with additional technology and service support. If you turn the vision into high-level future scenarios, you have a powerful communication package to guide your developers and stakeholders.

Storyboarding—Working Out the Details

Focused Rapid CD, which collects and consolidated sequence models, works out the details of the vision with storyboarding techniques. Too often when people design, they break the existing practice because they jump from their big idea to low-level user interface and implementation design. As soon as designers start focusing on technology, technology and its problems become their central design concerns. How technology supports work or life practice is subordinated. This also happens when people design from idealized models of how processes work.

The steps and strategies of a practice, being tacit, are easy to overlook. If we are out in the field watching them, we can see them. This is what is captured in the sequence models. Storyboarding keeps the team honest and the design clean. Guided by the affinity diagram and consolidated sequences the vision is made real in storyboards. Storyboards ensure that the team does not overlook any intents and steps that are critical to the practice. Even if the practice is changed, we have to think through the details of how it will be changed to ensure that adoption is easy.

Storyboards are like freeze-frame movies of the new practice (see Fig. 49.4). Like storyboarding in film, the team draws step-by-step pictures of how people will perform activities in their new worlds. Storyboards include manual steps, rough user interface components, system activity and automation, and even documentation use.

Because they focus on practice redesign, storyboards prevent the design team from prematurely delving into too much detail. They are guided by customer data, and after each task has been thought through and sketched, the team reviews it to ensure that it remains true to the customer data. This does not mean that no invention happens, but that the team must account for the steps and other data elements of the affinity. They must look at them and make a conscious decision about how to handle them. They might change all the steps and even eliminate whole sequences; as long as people can still achieve their fundamental intents, the change will work. When teams forget or ignore the user's intent, the design is in trouble.

How is storyboarding different from building scenarios? The storyboard is fundamentally a future scenario guided by the vision and reined in by the data. Every system implies multiple storyboards of use. Just as a kitchen supports multiple scenarios of cooking (the quick hot dog, the Thanksgiving dinner, the breakfast for the toddler), a system supports multiple scenarios of use. A storyboard is a high-level use case.

Any vision implies multiple scenarios to fulfill the vision. No system should be built around one storyboard or one future scenario. Gathering a set of people together in a room to invent the future with future scenarios, returns them to design from the *I*. Where do the cases come from that the system must handle? If not from customer data, they must be generated by the people in the room. If they come from idealized process flows, they are probably unrealistically linear, ignoring the multiple strategies and complexities of real work processes. Good systems are not linear; they do not force or follow a single linear path. Driving scenarios from consolidated data ensures that different strategies for doing a task are considered and that the design supports the different routes that real people use to achieve their intent and the different roles that people play when completing activities.

By working with multiple teams, we have learned that the best system design comes from pictorial representations of the future practice to be designed. Text versions of a scenario or simple user interface drawings alone fail to consider all the dimensions of the new practice. Use case and object modelers are trained to use the text-intensive use case characterization of the redesigned work. We find that this ultimately focuses the team too much on system activities and business rules and not enough on necessary human processes and user interface function. User interface designers who are concerned primarily with the interaction design tend to overlook the system steps and overfocus on UI detail. They get stuck prematurely in detailed discussions of layout that will become irrelevant when the implications of multiple scenarios of use are worked through.

Storyboarding consolidated sequences ensures that you maintain the breadth of real activity without having to make up

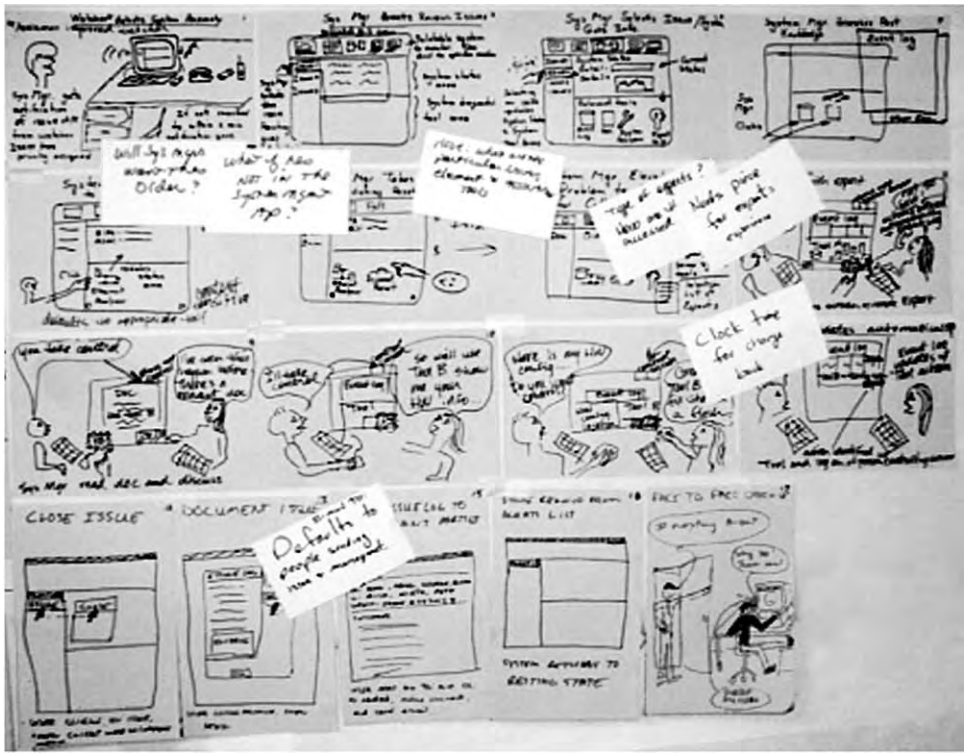


FIGURE 49.4. Storyboards are like freeze-frame movies.

cases. If the team is missing sequence data that supports the vision, they should collect it. Now, knowing what data is needed, data collection can be very focused. Get sequence data from two additional interviews and consolidate just those two sequences. If you can get only one sequence, analyze it as you would for sequence consolidation. Having a guide to the design is always better than making it up. Repeatedly, we have seen that teams trying to design future practice without data get lost and confused and are prone to overdesign. Data keeps teams focused and centered so invention is reasonable and likely to succeed.

The User Environment Design Option. For Rapid CD we recommend that you skip creating a formal User Environment Design (UED) and go straight to the user interface structure. For complex projects and systems, for doing a competitive analysis, or for analyzing a planned design to find potential problems, you should consider using the UED formalism.

A good product, system, or Web page must have the appropriate function and structure to support a natural workflow within it. System design really has three layers. The user interface accesses the function, structure, and flow necessary to support the user's redesigned work. The implementation makes that function, structure, and flow happen. The core of a product is that middle layer: the explicit work the system is performing. Just as architects draw floor plans to see the structure and flow of a house, designers need to see the floor plan of their new systems. Hidden within the storyboards are the implications for the system floor plan—the UED.

The UED formalism represents a set of focus areas or places in the system that provides support for coherent activities. A place might be a window, Web page, dialogue box, or pane. The UED shows each part of the system—how it supports the customer's work, exactly what function is available in that part, and how the customer gets to and from other parts of the system—without tying this structure to any particular user interface or implementation design. The function in the UED drives functional specification and implementation level use cases. The function in each focus area becomes the specification for the part of the user interface that will support that function.

In creating a UED, the team walks the storyboards and derives the implications of what the system needs to provide. As the implications of storyboard after storyboard are rolled into the UED, the team starts to see the best way to structure the system. This system structure now represents the system that, if built, will actualize the vision as it has been worked out in the storyboards. Whether this system will be valued by the users has yet to be tested. Any design team can anticipate needed function only up to a point. Therefore, after working out the preliminary UED, the team mocks up each focus area in paper and tests it in mock-up interviews. Through iteration with the users, the UED stabilizes, the user interface paradigm stabilizes, and the lower level requirements are solidified.

User Interface Design and Mockup

After visioning or developing a list of recommendations, the team will work out their first cut at how to present their redesign and new function to the users. In Lightning Fast+ the team uses their existing methods to come up with a first cut at a user interface. In Focused Rapid CD, the team works out the new practice and system in storyboards. Now the team needs to abstract out the elements of the design that will define the system and the user interface (UI) from the storyboards. Whichever process you choose the team will have to define how to lay out the needed function and information into a set of rough screens representing the core places in the system. These places in the user interface may augment an existing interface, add new places and features implied by the vision, or represent the changes that were worked out in more detail within the storyboards.

However, this function and initial layout is yet untested with users. To ensure that you have the right solution, testing and iteration is essential. Lightning Fast CD gives you the data you need to generate a concept but Lightning Fast+ and Focused CD ensure that you make the actual design and overall user experience work for the people who will use the system.

To test your design concepts and clarify your functions, we recommend constructing a paper prototype and testing it with your user population in paper prototype field interviews. Users do not understand models or even storyboards when presented with them. However, users can talk user interface talk or form factor talk for physical products. So to test ideas, we move quickly through storyboarding and laying out the rough user interface to get back to the users with our ideas represented as user interfaces.

Build the paper prototype using normal stationery supplies (Fig. 49.5). Card stock provides a stable background to simulate the screen. Post-its effectively simulate anything that might be moved during an interview, such as pull-down menus or buttons. Sample content should be put on a removable sheet so that users can replace it with their own real content during the interview. Web content should be laid out for example products or content types to provide a context for discussing the structure of the content and its layout. For hardware designs, use other kinds of props to simulate the hardware. The final prototype is rough and hand drawn, but it represents both the system's structure and its behavior.

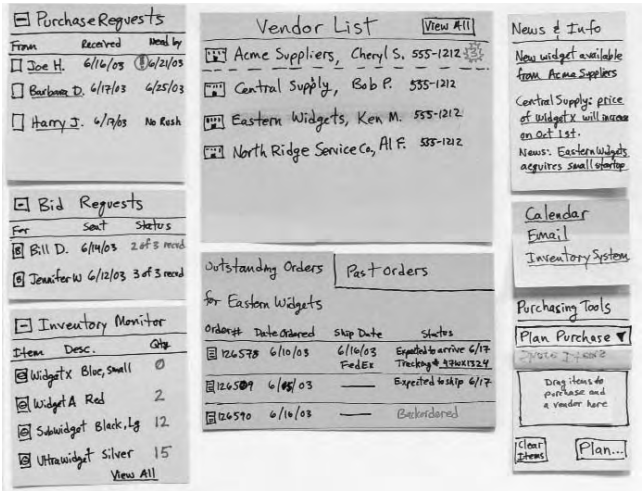


FIGURE 49.3. A complete, synthesized vision.

Mock-up interviews help designers understand why design elements work or fail and identify new function. These interviews are based on the principles of Contextual Inquiry given earlier. We test the paper prototype with users in their contexts to keep them grounded in their real practices. Users interact with the prototype by writing in their contents and manipulating and modifying the prototype. The partnership is one of codesign. As the user works with the prototype following a task they need to do or did in the recent past, the user and designer uncover problems and adjust the prototype to fix them. Together the user and interviewer interpret what is going on in the usage and come up with alternative designs.

After the design has been tested with three or four users, we redesign to reflect the feedback. Multiple rounds of interviews and iterations allow testing in increasing levels of detail, first addressing structural issues, then user interface theme and layout issues, and finally detailed user interaction issues.

If your time is short, you might want to take out your components for testing in stages. Start testing the central or critical components and grow the mock up over your rounds. Just be sure to test each component at least twice. You may need to add additional rounds if you test component by component. Be careful that you do test the system altogether or you might get good components that do not play well together.

Why build initial paper prototypes on paper—paper versus online testing. Designers like tools. We are so often asked why we use paper mock ups instead of online prototypes. We create mock ups in paper because they are fast to assemble and they allow the user and interviewer to make changes to the interface in the moment. No tool can assemble an interface in one or two days. More importantly, hand drawn paper prototypes make it clear to the user that icons, layout, and other interface details are not central the purpose of the interviews; it keeps the user focused on testing structure and function. The rougher the mock up, the easier it is to test structure and function.

As the interface is tested, it starts to stabilize. At round three of the mock-up interviews, we begin to solidify the interaction design and build our mock ups in wire frames. However, we still print and cut the wire frames up to allow for real codesign with the user and quick changes in the moment.

Toward the end of the prototype rounds, we use running prototypes to test low-level usability function that simply cannot be tested on paper; for example, how well the user can manipulate a handle for an object in a drawing tool. We also use high fidelity completed Web page designs displayed online to test alternative visual designs. This helps us determine optimal use of font, color, and spacing to promote scanning, to test various layouts for relationship and prominence of elements and flow from one area to another, to test the feel of navigating from place to place, to test the physical experience of using a device such as a cell phone, and to test those interactions that cannot be tested on paper.

However, these design reviews, as we call them, are not testing function; their intent is to test how well the visual design and layout support the design. Moreover, any of this testing comes after the mock-up testing has already solidified the system structure and function. To get the system right—the right function, structure, and flow—start with paper and keep the user involved in the testing. Then, once you know you have the right overall direction, you can move to wire frames and online prototypes.

How are mock-up interviews different from demos, rapid prototyping, or usability tests? The primary issue in answering this question is where did the design being tested come from? Any time you iterate an existing design, you can expect no more than a 15 to 20% change to the structure that is being presented. People will have an opinion of what you are displaying before them; it is much harder to find out that what you are showing them is not fundamentally valued. The question is what kind of data you can get in each context:

- *Demos at a user* conference or in a focus group assume that people are aware of their practices, that out of the context of real use they can predict and project what they would need or want, and that their knowledge of their activities is not tacit but known such that they can bring it to bear in a discussion. We do not find this to be true. Demos and focus group discussions are better tests of sales point than function. They measure excitement with the product and clarity of the sales story.
- *Rapid prototyping* allows users' tacit knowledge to shape and tune a system that is being presented. Since the context of the discussion is to try to tune and shape what is already designed, the fundamental structure of the system and its value is rarely challenged. Some designers or technologists decided that the presented function was needed and should be structured in a particular way. CD does rapid prototyping but only after systematically using customer data to produce the basic design to be tested. In this way, customer data shapes the whole of the design, not just the final testing.
- *Usability testing* can be another kind of rapid prototyping and as such suffers from the same challenges if the tested system is not originally designed from customer data. This is why usability professionals are at the forefront of pushing processes like CD into their organizations. They know that usability has to be built into the product during definition phase, not left to the end of the process. At some point, paper mock-up interviews are no longer good enough to test fine interaction with a tool. Lab testing is good for checking user interface manipulation on a running prototype. Even for these we recommend using Contextual Inquiry techniques in the lab if the product is unstable or taking it out on laptops for testing with users' real tasks.

INTEGRATING CONTEXTUAL TECHNIQUES INTO STANDARD PROCESSES

Some companies have software methodologies within which any user-centered design process must fit. In general, CD, specifically Rapid CD, can fit into any software or system development methodology. Most methodologies define a series of stages, each with deliverables and milestones. Few define specific ways

of gathering requirements so CD can easily be fit within the requirements step of these methodologies. Methodologies differ in the deliverables they require for their milestones. However, the natural data and design artifacts of CD easily can be used as inputs into any methodology's format.

In this section, we discuss how Rapid CD supports different methodologies.

General Corporate Methodologies

Corporate methodologies follow a set of typical steps outlined below. Focused Rapid CD can support any general corporate methodology.

- **Business case.** Define the marketing or work improvement rationale for building a new or revised product or system. Use any Rapid CD technique up to the vision step to characterize the market and generate product concepts that can then be investigated and validated with focus groups, surveys, and other market intelligence.
- **Requirements gathering.** Gather user needs and overall business needs to guide building the product or system. Rapid CD techniques will get the detailed data on issues and tasks needed to develop a system and generate system requirements at the level of the UI and the function. Some companies would be satisfied with requirements extracted from untested storyboards. We recommend that you finalize your requirements after you define and test your user interface.
- **Design.** Detailed design of the product or system, both the user interface and underlying technology, making sure it can be implemented. In this step, you create the implementation structure that will support the function and UI defined through your Lightning Fast+ or Focused Rapid CD processes. The key to this step is using paper prototypes to finalize the lower level requirements and interaction design. There is usually some negotiation with development at this point to fit implementation constraints. Any changes at the user-visible level should be checked and, where necessary, tested with users.
- **Implementation.** Code and test the user interface and the overall system. Here you test the running system first with running prototypes and then during Alpha and Beta releases using Contextual Inquiry to gather low level and tactical changes.

If you must comply with a corporate methodology such as that outlined above, choose Focused Rapid CD and map the steps to your deliverables. Use the data and artifacts of the CD process to inform the documents and specifications you need to make for your corporate processes. If you are tasked just with developing a business case, then Lightning Fast Rapid CD can get you started.

Rational Unified Process

The traditional Rational Unified Process (RUP) approach (see Cockburn, 2001) is very similar to (indeed, is an outgrowth of) the preceding corporate process. RUP phases include business modeling, requirements analysis, design, implementation, test, and deployment. Use the Focused Rapid CD process to support RUP as follows:

- **Business modeling.** Use the first half of Focused Rapid CD up to visioning. The goal is to develop an understanding of how the business operates now. This may be captured in business process diagrams or as-is use cases. Rapid CD can be a critical tool in discovering the root causes of problems and the informal ad hoc processes that make the business process work. Consolidated sequences are as-is use cases.
- **Requirements.** Use the second half of Focused Rapid CD to collect any additional task data needed to redesign the tasks to be supported by the new system. Use cases are the key representation of user behavior in the new system. Storyboards are the to-be models and feed use case development.
- **Analysis and design.** Use storyboarding from a Focused Rapid CD process to feed the standard RUP process to develop object models directly from use cases. Consider developing a UED. It can be an important way to capture the proposed system and its relationships.

Agile or Extreme Programming

Extreme programming (XP; see Beck, 2000) is one of several Agile programming methods—methods that focus developing your system in a series of simple iterations, each of which delivers value to the customer. Rather than spending a lot of time in up-front planning and modeling, the XP process works closely with customer stakeholders to define each iteration and then builds it, ships it to the customer for feedback, iterates it, and moves on to the next piece. The system is built up quickly through rapid iteration.

A key role on an XP team is that of the customer, who defines what needs to be done, prioritizes tasks, and works closely with the developers to work out each detail of the system. All the problems discussed earlier apply to this role: customers are not aware of their practices, no one person can represent all users, and people who no longer do the activities are not representative users. So use the Rapid CD process to give your customer role the right information to guide the team well.

XP organizes development around rapid iterations using release planning and iteration planning to decide what to build. User stories define how users will work in the new system. We recommend Rapid CD for XP, making it a more user-centered design process.

- **Release planning.** Rapid CD provides the detailed information to decide what should go into a release. The XP philosophy is not to plan too far ahead because things will change anyway; accordingly, use a Lightning Fast style project to raise key issues. Use Focused Rapid CD through storyboarding to get sequences to guide the development of user stories. Bring all data into the release planning meeting and work out with your customer what to implement for this release.
- **Iteration planning.** Use CD field interviews to track the success of each iteration with your actual users. Identify prob-

lems for each iteration as you go, providing continual refinement to your project direction. Use the iteration planning meetings to look at issues being raised by prior iterations, and agree with your customers on prioritizing solutions. Use paper prototyping to iterate your design with users in paper or wire frames before planning them into iteration.

Rapid CD maps reasonably well into any formal methodology. Rapid CD artifacts and data can easily supplement or feed deliverables required for any of the standard methodologies. Any methodology can use the consolidated customer data to drive brainstorming of recommendations, or the visioning process to generate a more systemic response to the data. Visioning is close enough to brainstorming, a known technique in any company, that it should be able to be easily adopted.

Similarly, paper prototyping has begun to be widely accepted as a means of testing design ideas and refining them. Although we recommend doing it in the field—with real user cases and not the canned test scenarios of some usability testing—paper prototyping as a technique of getting feedback from users is critical for working out the details of a particular design. Sometimes this is the best first step in getting user data into a design and letting developers and designers alike see how their systems are being received.

Rapid CD can also become the basis for the development of personae to augment the design process. Usability testing is built into the process such that good user experience is a natural outcome of any Rapid CD process. If companies just want to get started infusing user data into their processes, they can use a Lightning Fast Rapid CD. In only a few weeks, you can begin to understand your market or user population and generate solution concepts.

CONCLUSION

The core of Rapid CD is getting customer data into the minds of product managers, designers, and developers. It provides the data that is needed to guide business decisions, prioritize requirements, identify how to streamline work, be clear on what will be of value to the user, and produce high quality user experience. Becoming user centered is accessible today to any company that really wants to be user centered. Time is not an issue—being user centered is a business decision.

References

Beck, K. (2000). *Extreme programming explained: Embrace change*. San Francisco, CA: Addison-Wesley.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems* (1st ed.). San Francisco, CA: Morgan Kaufmann.

Cockburn, A. (2001). *Writing effective use cases*. Boston, MA: Addison-Wesley.

Cooper, A. (1997). *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity*. Indiana, IN: Sams Publishing.

Holtzblatt, K., Wendell, J. B, & Wood, S. (2005). *Rapid contextual design: A how-to guide to key techniques for user-centered design.* San Francisco, CA: Morgan Kaufmann.

Manning, H. (2003, December). The power of design personas. *Forrester Research*.

·50·

AN ETHNOGRAPHIC APPROACH TO DESIGN

Jeanette Blomberg
IBM *Almaden Research Center*

Mark Burrell
Microsoft Corporation

Introduction **966**
The Relevance of Ethnography for Design **966**
The Roots of Ethnography **967**
Principles of Ethnography **967**
Natural Settings 967
Holistic 967
Descriptive 967
Members' Point of View 968
The Postmodern Inflection **968**
Ethical Issues **968**
Ethnographic Methods **969**
Research Planning 969
Study Participants 969
Gaining Access 970
Observation 970
Why Observe? **970**
The Researcher's Observational Role **971**
Structuring Field Observations **971**
Videotaping **971**
Interviewing 971
Social Network Analysis **972**
The Interview as a Communicative Event **972**
Interviewing Rules of Thumb **972**
Connections Between Observation and Interviews . . . **973**
Self-Reporting Techniques 973
Diaries **973**
Visual Stories **973**
Weblogs (e.g., "Blogs") **973**
Remote "Virtual" Observation 974
Artifact Analysis 974
Recordkeeping 974
Qualitative and Quantitative Data 975
Ethnography in a Global Context 975
Shifting Design Focus **975**
Globally Distributed Interactions **975**
Making Ethnography Matter: Communicating and Applying Ethnographic Insights to Design **975**
Designing What? 976
Products **976**
Experiences **976**
Services **976**
Organizational Processes **976**
Business Strategies and Models **976**
Representations and Models 977
The Value and Function of Representations and Models **977**
Enhancing the Working Models of Developers **977**
Supporting Innovation **977**
Evaluating and Prioritizing Ideas **977**
Guiding Principles and Shared Reference Points . . . **978**
Types of Representations and Models **978**
Sample Representations and Models **978**
Experience Models **978**
Process Models **979**
Personas **979**
Scenarios **980**
Mock-ups and Prototypes **980**
Caveat Regarding Representations and Models **982**
Relation to Other Qualitative Approaches and Perspectives **982**
Ethnography in Action **982**
Case Study 1: Designing a Program and Website to Change Healthcare Behaviors 982
Case Study 2: Department of Highways 984
Conclusion **985**
References **986**

INTRODUCTION

In recent years, academic and professional researchers and designers working in the field of Human-Computer Interaction (HCI) have looked to ethnography to provide a perspective on relations between humans and the artifacts and solutions they design and use.[1] Within the field of HCI there are different views among researchers and practitioners on just what constitutes an ethnographic inquiry. For some, *ethnography* is simply a fashionable term for any form of qualitative research. For others it is less about method and more about the lens through which human activities are viewed. In this chapter we will attempt to position the ethnographic approach within historical and contemporary contexts, outline its guiding principles, detail the primary methods and techniques used in ethnographically informed design practice, and provide case examples of ethnography in action.

This chapter provides an introduction to ethnography, primarily as it relates to studies in HCI. We will touch only briefly on some of the more controversial topics current within the field of ethnographic research that have enlivened mainstream academic discourse in recent years. We will point the reader to books and articles where these topics are discussed in more detail. Our primary aims in this chapter are to provide academics and professionals in the field of HCI with a working understanding of ethnography, an appreciation for its value in designing new technologies and practices, and a discerning eye when it comes to reviewing and evaluating ethnographically informed design studies.

THE RELEVANCE OF ETHNOGRAPHY FOR DESIGN

The turn to ethnography as a resource for design can be traced back to the early 1980s when computer technologies were moving out of the research labs and engineering environments and into mainstream office settings, call centers, manufacturing floors, and educational institutions. There was the realization that the designers and developers of these technologies could no longer rely exclusively on their own experiences as a guide for the user requirements of these new systems.Instead, designers and developers needed a way to gain an understanding of the everyday realities of people working within these diverse settings (Blomberg, Giacomi, Mosher, & Swenton-Wall, 1991). In many organizations, market research groups were being asked to provide perspectives on the people and practices that made up these varied settings. However, the techniques most commonly used by market research groups at the time (e.g., attitude surveys, focus groups, telephone interviews, etc.) were not well suited for developing an actionable understanding of what people actually do day-to-day that could inform the design of new products and interactive solutions.

Anthropologists and other social scientists had long recognized that what people say and what they do can vary significantly, making reliance on surveys, focus groups, and telephone interviews insufficient for the task. Designers and developers needed a way to get a firsthand view of the on the ground realities—the "here and now"—of everyday life in these diverse settings. At this time in the early 1980s, social scientists working at the Xerox Palo Alto Research Center were beginning to explore ways of bringing insights from ethnographic research into a productive relationship with the design of new technologies (e.g., Blomberg, 1987, 1988, 1995; Suchman,1983; Suchman et al., 1999). Not long after, other research labs (e.g., Hewlett-Packard, Apple Computer, and NYNEX) followed suit (e.g., Nardi & Miller, 1990; Sachs, 1995). Today many industrial research and development labs in the United States have anthropologists and other social scientists with ethnographic expertise on staff (e.g., IBM, Intel, Kodak, Microsoft, Motorola, General Motors, and Xerox, to name but a few).

Ethnographically informed design practices also began to take hold in design firms and consulting companies during the early 1990s (e.g., IDEO, Fitch, and the Doblin group). These early explorations culminated in 1993 with the founding of E-Lab, a research-and-design company that distinguished itself from other design firms at the time by creating an equal partnership between research and design (Wasson, 2000). Ethnographic methods were at the center of E-Lab's research approach, with a commitment to base design recommendations on insights from ethnographic research (Robinson, 1994).

Furthermore, in the mid-1980s the growth in networked applications and devices, made possible through the availability of local area networks (LANs) and early Internet implementations, created awareness among designers and developers that they would need to focus beyond the support of single, isolated users interacting with information technologies. What would be needed was a way of exploring the information and communication practices of people interacting with one another, both face-to-face and through mediating technologies. Information technologies were increasingly becoming communication-and-collaboration technologies that consequently demanded an examination of social interaction across time and space. In response, a group of computer scientists, human-factors engineers, and social scientists, somewhat dissatisfied with the dominant perspectives within HCI at the time[2] founded the field of Computer Supported Cooperative Work (CSCW) (e.g., Grief, 1988; Schmidt & Bannon, 1992.). A group of sociologists at Lancaster University and researchers at the Xerox Research Center in Cambridge, England played a prominent role in helping to shape the ethnographic research agenda within CSCW (e.g., Bentley et al., 1992; Hughes,

[1]Ethnographic research is often just one of many approaches used to inform design. Usability studies, surveys, business case analysis, scenario planning, future workshops and social network analysis are a few of the approaches that are used in conjunctions with ethnography.

[2]The dominant perspectives at the time emphasized technological possibilities over the uses and users of technology, the interface requirements of standalone applications over networked devices, and human psychology and cognition over social interaction. However, by the late 1990s ethnographically informed design attained a prominent place in HCI research and today there is considerable overlap between the fields of CSCW and HCI.

Randall & Shapiro, 1993; Rodden & Anderson 1994; Hughes, Rodden, & Anderson, 1995).

Finally, the explosion of the Internet in the late 1990s accelerated the move of information technologies out of the workplace and into homes, recreational environments and other non-work-related settings. This has redoubled interest in the ethnographic perspective as a valuable tool in the design of new technologies and technology mediated services. This has presented a new set of challenges for designers as they were asked to design and build applications that leveraged powerful, digital technologies for use by people of all ages, engaged in myriad nonwork related activities in diverse contexts. Although the clamor for all that is the Internet has somewhat subsided, the legacy of that period is that researchers and designers who learned their craft during the Internet boom years have gone on to positions in academia and industry, in both boutique design firms and major companies, and in a variety of industries including advertising, marketing, product development, and IT services. In late 2005 many in the ethnographic design community assembled at an industry sponsored conference, EPIC (Ethnographic Praxis in Industry Conference). The conference brought together a diverse group of researchers working in areas such as product design, workplace studies, business ethnography to define the scope of a collective agenda and to strengthen professional ties and research connections (Anderson & Lovejoy, 2005). This conference was a powerful testament to the continuing value of focusing on people's everyday realities and experiences—the here and now—when designing innovative technologies, experiences, and services.

THE ROOTS OF ETHNOGRAPHY

Ethnography has its historical roots in anthropology, but today is an approach found in most all of the traditional and applied social sciences, and in interdisciplinary fields such as Human-Computer Interaction and Human Factors Engineering. In anthropology, ethnography developed as way to explore the everyday realities of people living in small-scale, nonWestern societies and to make understandings of those realities available to others. The approach relied on the ability of all humans to figure out what's going on through participation in social life. The techniques of ethnography bear a close resemblance to the routine ways people make sense of the world in everyday life (e.g., by observing what others do, participating in activities, and talking with others). The research techniques and strategies of ethnography developed and evolved over the years to provide ways for the ethnographer to "be present" for the mundane, the exceptional, and the extraordinary events in people's lives.

Over the years within the field of anthropology both the focus on nonWestern peoples and the implicit assumptions made about non-Western societies (e.g., that they are bounded, closed, and somewhat static) have changed. Today, the ethnographic approach is not limited to investigations of small-scale societies, but instead is applied to the study of people and social groups in specific settings within large industrialized societies, such as workplaces, senior centers, and schools, and specific activities such as leisure travel, financial investing, teaching, and energy consumption to name but a few. Consequently, new techniques and perspectives have been developed and incorporated into anthropology and ethnographic inquiry. However, a few basic principles discussed later have continued to inform and guide ethnographic practice.

PRINCIPLES OF ETHNOGRAPHY

Natural Settings

Ethnography is anchored in the underlying assumption that to gain an understanding of a world you know little about you must encounter it firsthand. As such, ethnographic studies always include gathering information in the settings in which the activities of interest normally occur. This does not mean that ethnographic studies never involve techniques that remove people from those everyday settings or that introduce artifacts or activities that would not be present otherwise. The insistence on studying activities in their everyday settings is motivated by the recognition that people have only a limited ability to describe what they do and how they do it without access to the social and material aspects of their environments. Furthermore, the ability to fully articulate what we do is limited due to the tacit nature of the principles that guide our actions (Polanyi, 1966). Finally, some aspects of people's experiences can only be studied by observing and recording the ongoing flow of activities as they occur (e.g., people's patterned movements through settings such as retail stores or airports, moment-by-moment shifts in scheduling, etc.).

Holistic

Related to the emphasis on natural settings is the view that activities must be understood within the larger context in which they occur. Historically within anthropology the notion of holism focused attention on the fact that societies were more than the sum of their parts (however these parts were specified). The particular aspects of a society (e.g., the court system) could only be understood in relation to the other aspects of the society (e.g., kinship system, belief systems). Today, because ethnography is less often applied to the study of entire societies, the notion of holism has a somewhat different emphasis. Holism holds that studying an activity in isolation, without reference to the other activities with which it is connected in time and space, provides only a limited and potentially misleading understanding of that activity. So, for example, it would be of dubious value to investigate online search strategies without understanding how these strategies fit into the larger set of activities in which search is but one component (e.g., in the context of online trading, shopping, or report writing).

Descriptive

Ethnographic accounts have always provided a descriptive understanding of people's everyday activities. Ethnographers are

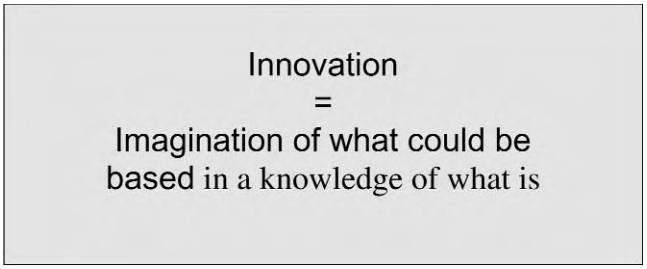


FIGURE 50.1. Innovation.

concerned first and foremost with understanding events and activities as they occur, without evaluating the efficacy of people's everyday practices. This is not to say that ethnographic accounts cannot or should not be used to suggest how things could be different or to point out inequities in current ways of doing things. Indeed, as applied in the domain of human-computer interaction, ethnography is often aimed at identifying opportunities for enhancing experiences. However, there is a strong conviction that to suggest changes or to evaluate a situation, one first needs to understand it as it is. The work practice and technology group at the Xerox PARC developed a slogan to express this conviction that innovation requires an understanding of the present (Fig. 50.1).

As such, ethnographic accounts strive first and foremost to provide descriptive and not prescriptive understandings of people's everyday lives. In recent years there have been many challenges to the idea that a purely descriptive understanding is possible. Critics point out that every account is shaped by the perspective of the researcher, the goals of the project, and the dynamics of the relationship between the investigator and those studied, to name but a few factors that shape ethnographic accounts. While it is hard to argue with this position, in our view the value of ethnography for design is not diminished by the recognition that our accounts are always located and partial.

Members' Point of View

As already alluded to, ethnographers are interested in gaining an insider's view of a situation. They want to see the world from the perspective of the people studied and describe behaviors in terms relevant and meaningful to the study participants. As such, ethnographers are interested in the ways people categorize their world and in the specific language people use to talk about things. This perspective is sometimes at odds with the requirements of quantitative survey research in which the relevant categories must be known in advance of the study and in which the categories and the language used cannot vary across participant groups. In such a quantitative categorical approach, the terms and categories used are likely to be those of the research community and not those of the study participants, which can undermine the validity of the results (see the section on ethnographic methods for further discussion of this topic).

THE POSTMODERN INFLECTION

The scientific paradigm within which ethnography evolved has come under serious questioning over the last quarter-century as social studies of science have shown how scientific knowledge production is shaped by the larger social context in which scientific inquiries take place (Latour, 1987; Latour & Woolgar, 1986; Pickering, 1980). As part of this critical discourse, ethnographic accounts have been challenged for their veracity. Likewise the authority of the ethnographic voice has been questioned (Clifford, 1988; Clifford & Marcus, 1986, Marcus & Fischer, 1986). These challenges have come from a number of fronts, most significantly from study participants who increasingly are able to read ethnographic accounts (Said, 1978) and from feminists who saw in many ethnographic accounts a Western, male bias (Harding, 1986; Smith, 1987; Wolf, 1992; Yanagisako & Delaney, 1995). These challenges have made researchers from all fields of inquiry more aware of how their research is shaped by the particular time and place in which it occurs. It is our view that knowledge of the world is always mediated by presuppositions, be they cultural, theoretical, or practical, and as such no ethnographic account is value-free. But we also contend that this does not diminish the value and efficacy of an ethnographic approach as a resource for designing new technologies, experiences, and services. Maintaining the illusion of a theoretically neutral and value-free absolute "truth" is not necessary to establish the efficacy of ethnographic research in design. By striving to describe and understand how people operate in and construe their everyday "realities," ethnography can provide useful frameworks and roadmaps to guide the design of "people-centered" solutions.

ETHICAL ISSUES

As will be discussed in more detail later, ethnographic research requires developing the trust and participation of the people studied. Without this trust participants will be reluctant to allow researchers into their homes, boardrooms, and classrooms, and they will not openly share their everyday experiences and concerns. Anthropologists have long realized that such a privileged, trusted position requires reciprocity—if you allow me access to your world, I will protect your interests. This bargain has not always been easy for ethnographers to keep. Over the years there have been examples of ethnographic research, where, wittingly or not, the situation of the people studied has been compromised.[3]

In the context in which ethnographic research is being used to inform the design of new technologies—technologies that

[3]To mitigate such negative impacts the American Anthropological Association has developed a code of ethics that provides guidance for people engaged in ethnographic research. This code outlines the appropriate disclosures and protections that should be given to study participants. (Fluehr-Lobban (1991) provides a discussion of ethical issues in anthropological research.)

will change people's lives—it is critical that the ethnographer reflect on the impact this research could have on study participants. Of course, it is not possible to control all the ways findings from ethnographic research will be used, nor how technologies informed by these studies will be integrated into people's lives. But the ethnographer can work to protect study participants from immediate harm (e.g., that caused by divulging a worker's identity to management) and can inform study participants of possible longer-term negative impacts (e.g., job losses brought about by introduction of new technologies). As ethnographic research has moved into new contexts (e.g., HCI, organizational development), it has been necessary to think creatively about how our ethical guidelines map to these new conditions. However, we cannot lose sight of the importance of protecting the interests of those who have agreed to participate in our studies be they workers in organizations, traders on Wall Street, or mothers of special needs children.

ETHNOGRAPHIC METHODS

The ethnographic method is not simply a toolbox of techniques, but a way of looking at a problem, a "theoretically informed practice" (Comaroff & Comaroff, 1992, quoted in Agar, 1996:7). The methods and techniques outlined later in this chapter have been developed over the years to enable the development of a descriptive and holistic view of activities as they occur in their everyday setting from the point of view of study participants. We are not attempting to be exhaustive in our presentation, nor do we want to suggest that there is a fixed set of canonical ethnographic methods and techniques. We encourage researchers to continue developing new techniques as the circumstances require (e.g., studying "virtual" communities, globally distributed work groups, technologically mediated interactions). What we believe remains constant in the ethnographic approach is a commitment to describe the everyday experiences of people as they occur.

Research Planning

One of the keys to a successful research project is the creation of a plan of action to guide the research and support changes and adjustments that inevitably must be made as the project proceeds. Research planning can be divided into three general stages: formulating research objectives, devising a strategy for selecting study participants, and selecting appropriate research techniques and approaches.

Research objectives follow from the specific questions to be addressed by the research. It can be useful to develop an explicit statement that clearly articulates the objectives of a given study. This statement acts as a beacon to help keep the research on track through the many twists and turns of a project. For example, if the research aims to inform the development of a software application that will help doctors manage patients' records, the research statement could be something as simple as "understand how doctors manage patient records through all the phases of treatment and in the varied settings in which they practice medicine." Over the course of a project, the research objectives statement (along with the research design and plan) may change as a project team coalesces and learns about the experiences of the people in the particular domain of interest.

Study Participants

Once the research objectives have been identified, a strategy for selecting study participants (sometimes referred to as a "sampling strategy") is devised that answers two primary questions: what types of participants best suit the research objectives, and how many participants should be included in the study to achieve the research objectives? The strategy for selecting study participants is influenced by the research focus (e.g., shopping behavior vs. work-group collaboration) and may include selecting at different levels of abstraction (e.g., which organizations, which workgroups, and which individual employees). In addition, as Cohen (2005) has cautioned, we should be attentive in making these choices for those who we intentionally or inadvertently excluded from our studies and as such are rendered invisible to our research lens.

Several types of sampling strategies are employed by social science researchers, which fall under two main categories: probability and nonprobability (Bernard, 1995).[4] Our focus in this chapter is on nonprobability sampling, as that is the most commonly employed in ethnographic research.[5] The nature of ethnographic work, as well as recruiting constraints often demand selecting participants based on criteria other than a strict probability.

Four types of sampling fall under the rubric of nonprobability: quota, purposive, convenience, and snowball (Bernard, 1995). When sampling by quota, the researcher specifies which groups are of interest (e.g., women, teenagers, truck drivers, people who use software X, organizations with fewer than 100 employees, etc.) and how many will be needed in each group. The number of groups chosen will depend on the research objectives and the amount of time available, but the basic idea is to cover the range of possible variation one would expect across the target population. Practically speaking, when identifying the variables or factors that should be considered in sampling to enable visibility into possible variations in experiences and practices,

[4]The intent behind probability sampling, or statistical sampling, is to generalize from the research sample to a larger population with a specified degree of accuracy, measured in terms of probability. All types of probability sampling require a randomly selected and relatively large sample size.

[5]Using non-probability samples does not mean we cannot make general statements. If participants are chosen carefully, one can obtain reliable data with as few as four or five participants (Nielson & Landauer 1993, Romney et al., 1987). Additionally, a recent case study demonstrates that smaller, non-randomly selected samples can produce the same results as large-scale survey research for as little as 1/100 of the cost (Green, 2001). A nonprobability strategy also does not preclude conducting a statistical analysis or measuring differences between individuals or groups using nonparametric statistics, such as Fisher's Exact Test or nonparametric correlation measures. Their limitation is that they cannot be used to make claims about larger user populations within a specified degree of probability.

the ethnographer will often presumptively identify "differences that may make a difference" in the experiential domain of inquiry. For example, if the focus is on how people think about and manage their personal finances, the researcher might deliberately strive to specifically sample people with varied financial situations in addition to life/career stages or family situations. To ensure the desired variability is covered—particularly when the researcher is dependent on others to provide access to or recruit the participants for a study—it is useful to create a "screener",[6] a questionnaire-like instrument designed to identify characteristics that are appropriate for a given project. Quota sampling is only possible when the desired participants[7] are easy to identify in advance and recruit. If it is not possible or desirable to specify how many participants will be in each sampled group, a purposive sampling strategy may be called for. This sampling strategy is based on the same principles as quota sampling, but the number of participants for each group is not specified.

Convenience and snowball sampling rely on a "sample as you go" strategy. This is required in situations in which you don't know in advance who will be available to participate or which individuals or groups should participate. Convenience sampling entails selecting people who are available, meet the requirements of the research, and are willing to participate. One might use this strategy, for example, to observe and interview people as they shop in a grocery store.

Snowball sampling relies on participants referring others whom they think would be good candidates for the research, or on researchers identifying individuals or groups to be in included in the study as the research proceeds. Because this method utilizes existing social networks it is especially valuable when desired participants are initially inaccessible or reluctant to participate (e.g., CEO's, drug users, club members) or when the relevant population cannot be known in advance.[8]

Gaining Access

One of the challenges for ethnographic research is gaining access to field sites and study participants. Access to institutional settings often requires getting permission from management to observe and interview employees, or from school officials and parents to spend time in classrooms. In some cases, written permission that specifies certain terms and conditions (e.g., how confidential information will be protected) is required before researchers are allowed onsite. In other cases, recruiting agencies may be used to identify participants and financial incentives may be offered for participating in the study. The time (and skill) required to establish these initial relationships and agreements should not be underestimated.[9]

Observation

As discussed earlier, ethnographers are interested in understanding human behavior in the contexts in which it naturally occurs, making observation one of the hallmark methods of the approach. In academic settings, it has been common for anthropologists to spend a full year at a given field site. While this continues to be the case for more traditional ethnographic studies, shifts in research focus (e.g., away from studies of entire societies), and in study locations (e.g., away from isolated, hard to reach settings) have resulted in more varied research designs which may involve shorter, intermittent fieldwork periods in one or more distributed locations. Moreover, in some applied settings (e.g., enterprise work environments) the time available for field observation may be constrained, sometimes allowing for no more than a few days in any one setting.

WHY OBSERVE?

One of the fundamental axioms in the social sciences, and anthropology in particular, is that what people say they do and what they actually do are frequently quite different. Studies have shown verbal reports to be inconsistent with observed behavior in a number of areas, including (among many other examples) shopping behavior (Rathje & Murphy, 1991), child rearing (Whiting & Whiting, 1970), recycling (Corral-Verduga, 1997), and health habits (Rich, Lamolu, Amory, & Schneider, 2000).

The discrepancies between verbal reports and behavior can be due to a variety of factors. People may be concerned with their image and so report, consciously or not, behavior that is more socially acceptable. Along these same lines, participants may respond to questions in a particular way in an attempt to please the researcher. Another source of disparity between behavior and verbal reports is that people are often not aware of their actual behavior because it is so habitual. Such tacit knowledge is often not easily accessible through interview techniques alone (D'Andrade, 1995).

The limitation of human memory is another reason why interview data can differ from observations. When asking participants about past events, or recurring patterns of behavior, our memory may be selective and skew responses in any number of directions, sometimes in predictable patterns (Bernard, 1995).

The complexity of social life is another reason individual accounts of an event may miss certain relevant details. The environments in which humans interact are extremely dynamic and complex—composed of social relationships, artifacts, and physical spaces—can make it difficult for individuals to fully envision, let alone articulate after the fact, what is going on.

---

[6]Screeners are an essential tool if using an external recruiting agency to locate study participants.

[7]For sampling purposes participants need not be individuals, but could be families, households, work groups or other naturally occurring entities.

[8]Johnson (1990) provides a more detailed discussion of sampling in ethnography.

[9]Anthropologists have been accused in the past of only studying the disempowered and disenfranchised because these individuals were less likely to feel powerful enough to refuse participation in ethnographic studies. Although important in all contexts, when studying people with more power and ability to say no (Nader, 1974), it is often necessary to demonstrate how their participation will be of benefit to them, their community or workplace, or the wider society.

THE RESEARCHER'S OBSERVATIONAL ROLE

When it comes to observation, there are varying degrees to which the researcher can become integrated into the scene. At one end of the spectrum the researcher may become an *observer-participant*. In this role, one attempts to be as unobtrusive as possible, quietly observing events from a discreet, yet strategic, position. At the other end of the spectrum is the *participant-observer*. In this situation, the researcher is actively involved in the events observed (e.g., a researcher who goes through the training to be a machine operator in an industrial environment).

There are pros and cons associated with each type of role. While being fully integrated into the action provides a researcher with firsthand experience of an event, taking good notes in this context is difficult at best. A great deal of energy is spent trying to fit in rather than on attempting to make sense of the events in the context of the research objectives. In such cases, one must rely on memory of the events when writing up field notes after the fact. Taking a more observational role affords a wider perspective on events and the time to record and reflect on events as they unfold. On the downside, it precludes the opportunity to experience the activity firsthand. In many research situations, the ethnographer's position moves between these two extremes, sometimes occupying a hybrid position of both partial participant and outside observer.

STRUCTURING FIELD OBSERVATIONS

Before setting out to observe, decisions need to be made about what, where and when to observe (Whiting & Whiting, 1970). One might decide to observe individuals as they go about their work and daily routines (person focused), a technique sometimes referred to as "shadowing" (Wasson, 2000). The researcher might also decide to focus on a specific event, such as a meeting or software education class (event focused), or observe the activities that occur over time in a given area, like an office or store (place focused). One can even shift the subject of observation to an artifact, such as a document, and record its transformation as it moves from person to person or along a development path (object focused).

VIDEOTAPING

Given the complexity of human behavior it is impossible to notice and record in real time everything of interest to the researcher. This is one reason video cameras have become increasingly popular in fieldwork. Video records can be used as a reference to supplement field notes. The ethnographer also has the advantage of being able to watch events multiple times and change levels of analysis or observational focus with subsequent viewings (e.g., interaction between people vs. the movement of one individual in and out of a scene).

Videotaping also allows people not primarily involved in the fieldwork to participate in the analysis and opens up the range of perspectives that can be bought to bear on the analysis (e.g., Blomberg & Trigg 2000) used video collection tapes in interactions with product developers; also see Brun-Cotton & Wall, 1995; Karasti, 2001; Suchman & Trigg, 1991).

Video cameras can also be used to record events in the absence of the researcher. Not only does this free the researcher to be involved in other activities, but the camera also can be a silent presence[10] in situations where an outsider (even a well trained participant observer) would be seen as intrusive (e.g., child birth, counselor-student interactions, board room deliberations, etc.). This however requires devoting time later to review videotapes and incorporate relevant information into the analysis.[11]

Interviewing

Interviewing is a central tool of ethnographic research (Gubrium & Holstein, 2002). Conducted and interpreted in light of the potential differences between what people say and do, interviews are critical in developing understandings of members' perspectives. Interviews can be placed on a continuum from unstructured to structured, with at one extreme the casual conversation and at the other a formal structured interview.

Ethnographic interviews are most often open-ended, particularly during the early stages of fieldwork when the ethnographer is just beginning to get a perspective on the activities and people studied. The more unstructured format gives the researcher the freedom to alter the line of questioning as the interview unfolds. The researcher essentially is learning what questions are important to ask. Unstructured, however, does not mean haphazard or lacking purpose. The researcher will know the research objectives and the topics to be explored when entering the field, and will usually have an interview protocol to serve as a (flexible) guide for the interview. While the protocol provides a basic framework for an unstructured interview, the participant plays a major role in the direction the interview takes. As Bernard (1995) wrote, the idea is to "get an informant on to a topic of interest and get out of the way." When the interview moves to a topic of particular interest, the researcher can then probe deeper to elicit more details. Indeed, interviewing is something of an art, and one of the key skills an ethnographer learns is the art of "interrupting gracefully" (Whyte, 1960).

In an open-ended interview it is important to avoid using an interrogation style of questioning (e.g., "yes or no" questions) which is designed to uncover the "facts." This defeats the purpose

[10]However, the expressed permission of the participants in the interaction is needed in these cases as well.

[11]A variety of software applications now exist which can help the researcher manage and analyze recorded on video. Caveat, for example, allows the researcher to select and annotate images/events of particular interest. A more sophisticated (though less user friendly) program is Observational Coding System (OCS) which provides for a more quantitative analysis.

of keeping the interview open to allow for a wide range of responses and for the participant to express his experiences, his own way, with his own words. Using too structured a format constrains the range of possible answers, increases the chances of missing critical pieces of information, and increases the risk that discoveries will be limited by the ethnographers' preexisting concepts, assumptions, and hypotheses. It is critical to provide opportunities for participants to convey their stories and perspectives in their own way and for the researcher to be surprised by what people say and do.

As a project progresses and patterns begin to emerge, interviews can become more structured and the line of questioning less broad. The researcher begins to narrow in on topics that are particularly informative and relevant to the research objectives. Questions on the protocol become more focused and specific as answers to previous questions guide the follow-up questioning.

Once the range of responses is known and the data begins to show patterns and themes, the researcher may want to structure interviews further. A host of structured techniques exist. Some are designed to identify the ways people organize information within a specified domain, such as free listing, card sorts, triad's tests, and paired comparisons (Romney, Batchelder, & Weller, 1986; Weller & Romney, 1988). Other techniques, such as questionnaires and surveys,[12] are used to assess variations between two or more groups or to establish how representative the findings are for a larger population. The main idea behind these techniques is to keep the form and content of the questions consistent for each respondent, thus allowing for differences among the sample population to be ascertained. Conducting structured interviews at the end of an ethnographic study has the advantage of allowing the question structure and language to reflect the way participants talk about and organize experiences, thus increasing the validity of the survey findings.

SOCIAL NETWORK ANALYSIS

A research technique that has received a great deal of attention lately is social network analysis (Cross & Parker, 2004; Kilduff & Tsai, 2003; Scott 2000). While the data for social network analysis (SNA) can derive from records of activities (e.g., e-mail message exchanges, coauthorship, membership in organizations), often the data used in SNA analysis are based on responses to survey questions (e.g., How frequently do you interact via e-mail with the following people?). However, interpreting the results of SNA often relies on other sources of information (e.g., ethnographic research). For example, although a SNA may show that there is little communication between two groups of workers, it cannot reveal whether this communication pattern is limiting the effectiveness of the two groups or is an indication that there is little need for the two groups to interact. Interpreting the patterns highlighted in social network analysis requires other ways of gaining an understanding organizational dynamics.

THE INTERVIEW AS A COMMUNICATIVE EVENT

The interview has become somewhat ubiquitous in western societies and is viewed as a reliable means of acquiring information of all kinds (e.g., attitudes toward tax increases, the value placed on education, preferences for certain products, basic demographic data, etc.). However, as Briggs (1983) pointed out, what is said in an interview should not be thought of as "a reflection of what is 'out there'" but instead must be viewed "as an interpretation which is jointly produced by the interviewer and respondent." This view compels us to regard the interview as a communicative event in which the structure and context of the interaction conditions what the researcher learns. This is no less the case in highly structured interviews (see Jordan & Suchman (1990) and Moore (2004) for a critical analysis of the ecological validity of survey research). Briggs recommends that we adopt a wider range of communicative styles in our interactions with study participants, particularly styles that are indigenous to the study population.

INTERVIEWING RULES OF THUMB

While there are no hard and fast rules for interviewing, a few general guidelines will help facilitate the interview process and increase the chances of obtaining useful information. There are some points to remember:

- Interview people in everyday, familiar settings. Not only does this make the participants more comfortable, it allows them to reference artifacts in the environment that play an integral part in their activities. Moreover, a familiar environment is full of perceptual cues that can help jog the not-so-perfect human memory.
- Establish and maintain good rapport with participants, even if it slows the interview process.
- Don't underestimate the value of casual conversation. Some of the most insightful information comes from informal conversations when social barriers are lowered.
- Assume the respondent is the expert and the researcher the apprentice. This not only shows the participant respect, but also gives them confidence and facilitates conversation. Even if the interviewer happens to be more knowledgeable on a particular subject, the goal of an ethnographic interview is to understand the participant's perspective.
- Use lack of knowledge as a discovery tool. Participants will always know more about their own experiences than the interviewer will. In this context, don't interrupt unnecessarily, complete a participant's sentences, or answer the questions. Again, the idea is to learn about the respondent's point of view, not the researcher's. In this context, the researcher's "inevitable ignorance" about the experiences of another person can be a powerful tool.

[12]A good introductory book on surveys is How to Conduct Your Own Survey (Salant & Dillman 1994). Readers interested in a more advanced treatment of the subject are referred to Babbie (1990).

- When conducting an open-ended interview, avoid asking "yes or no" questions. Responses to these questions provide less information than questions beginning with "what" or "how."
- Be flexible enough to adapt the line of questioning when necessary. Human behavior is complex and full of surprises.

CONNECTIONS BETWEEN OBSERVATION AND INTERVIEWS

As noted earlier, one of the defining qualities of ethnography is its emphasis on holism. To obtain this holistic view, combining different sources of data is useful (Agar, 1996). Observation alone is seldom enough to adequately address research objectives. As such, observation is invariably coupled with interviewing. Interviews can extend and deepen one's understanding of what has already been observed. Similarly, interviews can be conducted prior to observing, giving the researcher a better idea about what is most appropriate to observe.

Interviews can also be conducted in the context of ongoing activities, sometimes referred to as "contextual" or "in situ" interviewing. Instead of setting aside a specific time and place for an interview, the researcher creates an opportunity to ask questions as participants go about their daily activities. The strategy can be extremely useful in getting answers to questions that are prompted by observation of ongoing activities.

Self-Reporting Techniques

In cases where the domain of interest transpires over a long period, or in which direct observation is not practically feasible, self-reporting techniques can be very valuable. This methodology is especially good at revealing patterns in behavior or obtaining data that is otherwise inaccessible (Whyte, 1984). A number of self-reporting techniques exist which vary in terms of form, focus, structure, and mechanism of self-reporting. Common techniques range from simple written diaries to visual storybooks, and more recently to Internet-based (and often multimedia) "blogs."

DIARIES

Traditional diaries consist of written records, which might include personal thoughts or descriptions of specific behaviors or accounts of events in which an individual participates. The focus, format, and degree of structure of diaries used in ethnographic research vary depending upon the research objectives, ranging from structured activity logs which invite the participant to capture and describe specific aspects of her experiences for each entry, to relatively unstructured forms in which diarists are provided only with general instructions. Study participants might be asked to keep diaries regarding the specific contexts, foci, modalities, and outcomes of their interactions or they might simply asked people to describe their experiences over time while using a specific product.

How diaries are analyzed depends on the research objectives and resource constraints. If time permits, follow-up discussions with participants to clarify points or gain a deeper understanding of the meaning behind the words can be useful. The texts can also be coded for themes, key words, or phrases and patterns examined across individuals or between groups.[13]

VISUAL STORIES

Visual stories are essentially pictorial diaries that employ images in addition to text in order to document experiences. They can be particularly valuable when working with non-literate participants, such as children, or in situations where words alone are inadequate to capture the essence of the subject (Johnson, Ironsmigh, Whitcher, Poteat, & Snow, 1997). Much like more traditional text-based diaries, visual diaries can be employed and structured in any of a number of ways. Wasson (2000), for example, described giving participants a written guide directing them to take photographs of their interaction with a product under study. They were then asked to organize the developed photos into a story that made sense to them, and researchers conducted follow-up interviews over the telephone.

A more open-ended framework can also be informative. Interested in cultural differences between Italian and American fishermen, Johnson and Griffith (1998) instructed participants from both groups to take photographs of whatever they wanted. After developing the film, Johnson coded the pictures based on their content and found significant thematic differences between the groups, which added to his understanding of differences in cultural values of the two groups of fishermen.

A more recent derivation of the visual story utilizes a video camera which allows the participant to provide a running narrative alongside the visual content. Being able to experience the two sources of information simultaneously provides the researcher with a rich record of an activity. Blomberg, Suchman, and Trigg (1996) used a video-story approach in their study of the document practices of lawyers. They set up a stationary video camera in the law office of a study participant and asked him to turn on the camera whenever he had occasion to retrieve documents from his file cabinet. The running narration recorded on videotape provided insights into the everyday use of the file cabinet that helped inform the design of an electronic file cabinet.

WEBLOGS (e.g., "BLOGS")

As Internet technology has evolved along with the accessibility and ease of digital media capture and online sharing, forms of self-reporting are evolving rapidly. Most recently, this is evidenced by the proliferation of Weblogs—better known as

[13]With varying degrees of success text analysis software have been used to help with large data sets. Some noteworthy programs: Ethnograph, NUD*IST, E-Z-Text, and NVivo.

"blogs"—in which a website is used to "post" online entries that may include textual narratives, digital photos, or digital video/audio. Although not developed specifically to support ethnographic inquiries, blogs can be a potentially very valuable research tool. Blogs may be particularly useful as a way for participants to self-report their use of online tools in the context of their online activities. Blogs also enable researchers to review participant "posts" as they occur as well as to engage in asynchronous online exchanges and dialogues with participant "bloggers". Indeed, as blogs increasingly are used in ethnographic research, they may blur the boundary between self-documentation and interviews, resulting in a blend of online self-reporting and intermittent online "conversations" via threaded participant/researcher posts.

Remote "Virtual" Observation

Continuing technological developments—in video, audio, wireless, network applications, tracking capabilities, and pervasive computing—have created new opportunities to "observe" and collect rich and dynamic information across geographies in real time as well as asynchronously. These technologies increasingly enable ethnographers to "virtually" observe in a wide variety of contexts. Using digital video and audio, people's behaviors can be tracked and analyzed as they interact with computer supported products and Internet-based networks.[14] Indeed, these technologies (along with the use of other digital tools such as blogs) enable what some have begun to refer to as "digital ethnography" (Masten & Plowman, 2003).

The pervasiveness of the "web cam" is perhaps the simplest illustration of how technology has expanded the observational capabilities of ethnographers. Internet-enabled digital video cameras can stream video in real time and can be remotely controlled. This digital video and audio can be viewed by multiple people across geographies either in real time or by accessing video archives. Such techniques and information sources can be particularly useful for geographically distributed research and design teams.

In addition, computer and online sensing, tracking, and analytic technologies that monitor, gather, collect, and integrate information on peoples' computer mediated activities can be a useful source of information for ethnographers. Although early tracking and analytic technologies required complex sifting and analysis of massive amounts of data to find meaningful nuggets, more recent tools enable sophisticated tracking of individual paths and activities as well as the ability to model online behavior. For example, "scenario"-based behavioral models (e.g., of online shopping, exploratory behavior, task completion, etc.) which define hypothesized patterns or sequences ("funnels") of online behavior can be used as an analytic lens to understand individual or group online behaviors. To date, these tools have been used primarily to measure aggregate completion of online tasks (e.g., online shopping, self-service) and to identify obstacles to user success (e.g., usability issues). However, over time and in conjunction with other sources of data and information they may become useful tools for ethnographers interested in patterns of online behavior and technology adoption. This may become particularly important as ethnographers attempt to understand the formation and interactions of distributed virtual communities (e.g., Rheingold, 2000).

The potential for using (and misusing) these sources of information will likely increase exponentially as pervasive computing increasingly enables the identification of (and response to) individuals across multiple physical and digital environments and the tracking of their activities. The collection and use of digitally enabled behavioral observations obviously needs to be carefully constrained by ethical considerations, particularly the respect for privacy and informed consent. In addition, as with any behavioral observation, it is critical to understand the context in order to interpret the meaning and significance of the behavior. In this respect, tracking computer-mediated behaviors by itself is insufficient and may simply result in the collection of massive amounts of relatively meaningless data. However, if used in conjunction with other sources of information (e.g., self-reports that illuminate peoples' intentions and meanings), patterns in digital behavior can illuminate aspects of behavior that are difficult or impossible for a human researcher to observe.

For example, it has been increasingly common for teams designing online services and tools to examine individual and aggregate patterns of online behavior (as reflected in web server logs or "client side" logs that are generated as a function of what users do online) to both identify usability issues as well as to examine patterns of technology, product, and service adoption over time (Kantner, 2001).

Artifact Analysis

Ethnographers have long had an interest in the material world of the people they study. The artifacts people make and use can tell us a great deal about how people live their lives.[15] Artifact analysis can be an important part of contemporary ethnographic studies (e.g., Rathje & Murphy, 1991). For example, conducting an artifact analysis of the stuff on people's desks can say a great deal about the people's work practices. Similarly, studying the contents of an automobile's "glove box" can tell a great deal about how the car is used. Depending on the kinds of research questions asked, it may be useful to include the collection and analysis of specific artifacts.

Recordkeeping

Although the authority of the ethnographic voice derives in part from the fact that the ethnographer is present and witness to

[14]The ability to virtually observe and track behaviors presents many ethical issues that cannot and should not be ignored. It is critical that ethnographers establish guidelines and protections if they engage in electronic, digitally enabled observations.

[15]Archaeologists rely almost exclusively on the artifacts that remain in archaeological sites for their interpretations of the behavior and social organization of past human societies.

events of interest, the ethnographer should not rely exclusively on experiential memory of these events. In all ethnographic research it is essential to keep good records. Field notes should be taken either during or soon after observing or interviewing. The specific nature of the notes will depend on the research questions addressed, the research methods used, and whether audio or video records supplement note taking. Field notes should at least include the date and time when the event or interview took place, the location, and who was present. Beyond that, notes can vary widely, but it is often useful to indicate differences between descriptions of what is observed, verbatim records of what is said, personal interpretations or reflections, and systematic indications of the flow of observed events and activities. When working with a team of researchers, field notes need to be understandable to other team members. This is often a good standard for the specificity of field notes even when working alone. If such a standard is maintained, it will be more likely that the notes will be useful to the researcher months and even years later, in the event reanalysis or a comparative study is undertaken.

Qualitative and Quantitative Data

In a previous section we touched upon the complementary nature of observational and interview techniques and the benefit of combining these two approaches. Triangulation of data can serve to connect quantitative and qualitative data as well. Sometimes, prior to the start of a project the only data available is quantitative, sometimes in the form of survey data focused on population characteristics. Qualitative data derived from ethnographic research can complement quantitative research by providing a meaningful context for interpreting the quantitative results. Qualitative techniques allow researchers to dig deeper after a survey has been tabulated, and aid in interpreting and explaining trends that the quantitative data might reveal (Guest, 2000). In addition, qualitative data can inform the content and language of more structured questions, thus making them more meaningful and relevant to the participants.

Ethnography in a Global Context

While ethnography has its roots in the study of small-scale, non-Western societies, the application of ethnography in the design of products and services has focused primarily on groups and individuals located in the developed regions of the world (e.g., North America and Europe). Two recent developments have led to a shift in the center of design activity. One is the emergence of the economies of less-developed countries, particularly Brazil, Russia, India, and China, the so-called BRIC countries, which are rapidly becoming major markets for products and services. An interest in serving these growing markets has led some firms to invest in designing products and services specifically for them by directly engaging designers, developers, and potential users from these developing regions (e.g., HP and Intel). The second development is the rapid increase in the use of Internet-enabled information technologies that connect workers, consumers, citizens, and organizations distributed around the world.

SHIFTING DESIGN FOCUS

As new markets open up around the globe, many businesses and organizations see an opportunity to create products and services specifically for these markets recognizing that the products and services suited for the developed West may not be appropriate for these other regions. As such these companies have established design initiatives focused on and located in countries like India and China. In some respects ethnography has come full circle in its application to design, contributing to understandings of the local contexts of people living in culturally and linguistically diverse settings (the sites in which ethnographic practice first developed). More than ever ethnographic principles and practices are applicable and necessary as the center of design activity moves outside the developed West.

GLOBALLY DISTRIBUTED INTERACTIONS

The methods and techniques of ethnographic research must contend with the increasing number interactions both at work and in domestic spheres that take place "virtually" between people separated in space and time. This is a challenge for ethnographic techniques that were developed to study communities of people who interact face-to-face. Our techniques and approaches must be adapted both practically and analytically to this new context where many more interactions are mediated by information technologies (e.g., instant messaging, blogs, e-mail, telephone, web conferencing, shared digital workspaces, and repositories) that transform traditional notions of place and real-time interactions. In many enterprises work teams are made up of people who are not co-located, many of whom are highly mobile in their work activities, requiring interactions to take place through conference calls, instant messaging, and e-mail. Furthermore, in some regions of the world, people travel significant distances for jobs and other opportunities. In these cases, interactions with friends and family, as well as with others living away from their native communities, are enabled by communication technologies (Horst & Miller, 2005; Green, Harvey, & Knox, 2005). Various strategies have been developed to study distributed, multisited groups including team ethnography (placing researchers in multiple locations), perspectival ethnography (focusing on the view from one of the local sites), and virtual observations (observing digitally mediated interactions).

MAKING ETHNOGRAPHY MATTER: COMMUNICATING AND APPLYING ETHNOGRAPHIC INSIGHTS TO DESIGN

This section outlines some of the ways in which the insights derived from ethnographic work can be represented and communicated in order to effectively inspire and guide the design of products and services. These ways of representing and communicating what is learned are intended as examples of how ethnographic work can be made relevant for design. However, before

we outline some of these representational forms and practices we should consider the possible foci of our design activities.

Designing What?

The application of ethnography to support a design agenda was directed initially toward informing the design of technologies, tools and products. However, more recently attention has expanded to include the use of ethnography to inform the design of experiences (Pine & Gilmore, 1999), services (Thomke, 2003), organizational processes, and business strategies and models. The establishment of a design school (the "d-school" as it is called) at Stanford University in 2005, dedicated to teaching "design thinking . . . to solve big problems in a human centered way," points to the expanded role ethnographic research can play in informing design, beyond the design of products. (http://www.stanford.edu/group/dschool/big_picture/our_vision.html.) Ethnographers are now involved in projects and contexts that span a range of problems from helping to design the next e-mail application, to designing tools to support system administrators, to new business models to reach small and medium businesses with IT services, to new customer services for retail banking. Moreover, the adoption of a technology is usually associated with changes in user experiences, individual work practices, and/or organizational models and processes which can also be within the scope of the design focus.

PRODUCTS

The application of ethnography to product design has received the most attention in the literature partly because many of the pioneers in the field worked in corporate research organizations of major technology companies (e.g., Xerox, Apple, and HP). In addition, early commercial applications of ethnographically informed design often focused on the design of consumer products, from cleaning products to automobiles to toys (Elab, Doblin group, Sonic Rim). It is not surprising therefore that many view product design, whether high-tech products like PDAs and online calendar applications or everyday consumer products like breakfast cereals and cold remedies, as the primary application of ethnographic research (Squires & Byrne, 2002).

EXPERIENCES

The publication of the Experience Economy (Pine & Gilmore, 1999) marked a shift in design focus to include the experiences that products and other artifacts enabled. Pine and Gilmore argued that the real challenge for businesses is creating engaging experiences for both consumers and corporate customers. The admonishment by a number of business gurus to pay more attention to the customer in the design of products also contributed to this expanded focus. Customers, it turned out, cared less about the products themselves and more about what the products enabled them to do or experience. Businesses became concerned with delivering quality experiences in which the products took on more of a supporting role. The canonical example often cited for this shift to an experience economy is Starbucks, where what is being sold is not simply a cup of coffee, but the experience of buying and consuming the coffee at Starbucks, including the elaborate choices available, the wireless access provided, the exclusive access to trendy music, and so on.

SERVICES

The service sector has come to dominate much of the world economy and increasingly new services are the site of significant change in the way we work and play (e.g., online dating services, GM's OnStar, business process outsourcing). While many innovative services are made possible by new technologies that provide the platforms[16] on which new service relationships are built, the value being exchanged is the service provided, not the technology per se. Ethnographically informed design strategies are now being applied to service design (Thomke, 2003).

ORGANIZATIONAL PROCESSES

Workflow systems have become ubiquitous within many organizations, orchestrating everything from employee travel-reimbursement processes to customer online-purchasing procedures. With this comes the opportunity to inform the design of these technology-enabled organizational processes through the study of existing work practices and processes. Here again the design focus is not so much on the underlying technologies (e.g., SAP, Siebel) that manage the workflow, but on the processes themselves. This is not to say that these studies will have no impact on the underlying technologies—for example, making them more flexible or end user configurable. But the design focus is on the workflow requirements, how people will interact with these systems and will be supported in executing processes.

BUSINESS STRATEGIES AND MODELS

Ethnographic research is also playing a role in the design of business strategies and models. Organizations are realizing that their competitive advantage is only partly related to the quality of their products and services. Equally important are the business strategies, including channels to the market, relationships with business partners, and the composition of employees. Many new business models have emerged in the last decade that capture new revenue streams such as advertising (e.g., Google, Yahoo), selling software as a service (e.g., salesforce.com), and facilitating networks of sellers and buyers or customers and providers (e.g.,

[16]See, for example, Service Oriented Architectures (SOA).

eBay, regional IT distributors). Ethnographic research can and is helping to inform these new business models.

Representations and Models

Whether the focus is on designing products, experiences, services, processes, or business strategies, the researcher must find ways to ensure that ethnographically derived insights effectively inform design innovations and decisions. Researchers can help make connections between ethnography and design in many ways. At the most basic level, this is achieved through active engagement, integration, and collaboration of researchers and designers.[17] Subsequent to conducting ethnographic inquiries, researchers can engage with design teams by acting as user proxies (e.g., helping to formulate and/or review design concepts in scenario-based reviews, providing feedback regarding relevant user expectations and behaviors as they relate to design concepts and decisions, etc.). Conversely, the active and direct involvement of designers in key elements of ethnographic fieldwork (e.g., participating in observations and interviews, collaborative analysis sessions, reviewing video and audio recordings and user artifacts, etc.) can enrich their understanding of the people who will interact with and use the solutions they design.

Although these forms of engagement are valuable, they limit the ability of teams to take full advantage of ethnographically derived understandings. They are restricted in the impact to the scope of the direct interactions between ethnographers and designers. This can be particularly limiting when designing multifaceted solutions, working with large and/or distributed design and development teams.

THE VALUE AND FUNCTION OF REPRESENTATIONS AND MODELS

To increase the value and impact of ethnographic research, explicit representations or models can be created which distill and communicate essential insights about people's experiences in forms that can be applied to design problems and decisions. Although the definition of *model* can be the subject of debate (as can the distinction between *representation* and *model*), for our purposes we are using the term to refer to explicit, simplified representation of how people organize and construct experiences and operate in relevant domains. The important point here is that well-constructed representations which communicate effectively can help connect everyday patterns of activity and experience with design solutions. More specifically, representations and models are tools that can serve a number of purposes including: enhancing the working models of designers/developers, supporting innovation and creativity, evaluating and prioritizing ideas and concepts, and providing guiding principles and shared reference points for design teams.

ENHANCING THE WORKING MODELS OF DEVELOPERS

In order to design a technology solution for people, designers must have at least an implicit working view of the people who will interact with the solution. Such working frameworks and perspectives may include assumptions about a range of essential characteristics of the people who will engage with the solution and the contexts in which they will do so (Newman, 1998). Indeed, some would argue that successful design requires a high degree of "empathy" with the target population (e.g., Leonard & Rayport, 1997; Koskinen, Battarbee, & Mattelmäki, 2005). Implicit and/or explicit assumptions or knowledge about "users" may be formed through some combination of direct experience (e.g., interacting with and/or observing people in the target population in controlled or noncontrolled settings) and secondary learning (talking with others about the target group, viewing videotapes of target activities, reading, analogy to other directly experienced groups, etc.). However formed, the working "models" of designers/developers may be of varying levels of complexity, robustness, coherence, consistency, and viability. The broad, deep, and contextualized understanding provided by ethnographic research can enrich the design team's implicit working models.

SUPPORTING INNOVATION

The design of technology solutions for people obviously poses a range of potential creative challenges at varying levels of complexity. What problems should be solved? What should be built? What kinds of experiences should the technology solution support or enable? What features and functions would be useful, compelling, and satisfying for a particular group of people in a particular domain/context? How can existing or emerging technological capabilities be used to enhance a particular group's, or to solve a particular problem? Even if there are clear parameters defining the functionality that will be built (e.g., a set of "requirements"), design teams must still generate a compelling, easy to use, useful, and satisfying way of delivering that functionality. By providing an understanding of the human domain (patterns of relationship, systems of meaning, organizational structure, guiding principles or rules, etc.), ethnography can promote creativity that matters (Robinson & Hackett, 1997)—relevant innovations that create new, realizable opportunities.

EVALUATING AND PRIORITIZING IDEAS

Design teams not only face the challenge of generating innovative ideas and concepts, but also the equally important task of evaluating and prioritizing ideas and options that arise from var-

[17]As noted earlier, the ethnographer should develop an understanding of the types of design decisions that the design team will need to make and a sense of what they need to know to inform those decisions.

ious sources (e.g., business stakeholders, end users, development teams). Although there are obviously many evaluative methods (e.g., scenario-based user testing, etc.), models derived from ethnographic research and analysis (e.g., scenario models, mental models, interaction/social network models, etc.) can provide a critical lens through which development teams can evaluate and prioritize ideas based on how they may fit into, not fit into, or change people's experiences. The need for evaluation and prioritization may occur at various points throughout the development process, ranging from decisions about features and functions, broad directions for design concepts, and so forth.

GUIDING PRINCIPLES AND SHARED REFERENCE POINTS

The learning derived from ethnographic analysis, particularly when represented as explicit representations and models, can serve as an experiential guidepost for individual designers and design teams throughout the development process. Even though these representations do not prescribe or specify what should be done, they can aid developers by focusing attention on essential aspects of an experience, highlighting variations in the experiences, and limiting exploration of experiential "dead ends." In other words, they can provide a general structure and direction within which a team can develop a shared understanding and focus its creative energies.

TYPES OF REPRESENTATIONS AND MODELS

Representations and models can vary, ranging from personas and scenarios to more abstract mental models and more complex work-practice models. The number, type, and form of models vary as a function of what is being designed, the audience, and the constraints on the design process. (e.g., K. Holtzblatt, Chapter X; Redish & Wixon Chapter X). For example, teams designing organizational tools may find it useful to model work environments and detailed task sequences; teams designing learning tools and programs may want to represent particular skill domains, as well as learning processes.

Practitioners have developed a variety of representations and models to inform the design-and-development process. For example, Beyer and Holtzblatt (1998) described a set of five work models (flow model, cultural model, sequence model, physical model, and artifact model) to reflect different aspects of a work domain. Pruitt and Grudin (2003) articulated the value (and risks) of personas to inform the design process, while Carroll (2000) described the value of scenarios.

SAMPLE REPRESENTATIONS AND MODELS

The varying scope, form, complexitym and function of different types of models are illustrated in following examples.

EXPERIENCE MODELS

The model presented in Fig. 50.2, is one of several developed in the context of ethnographic research and analysis for a financial services company serving individual investors. This company aimed to develop web applications that would facilitate customers' active engagement in the investment process with particular financial instruments. The model was intended to articulate and visualize a financial development process as well as

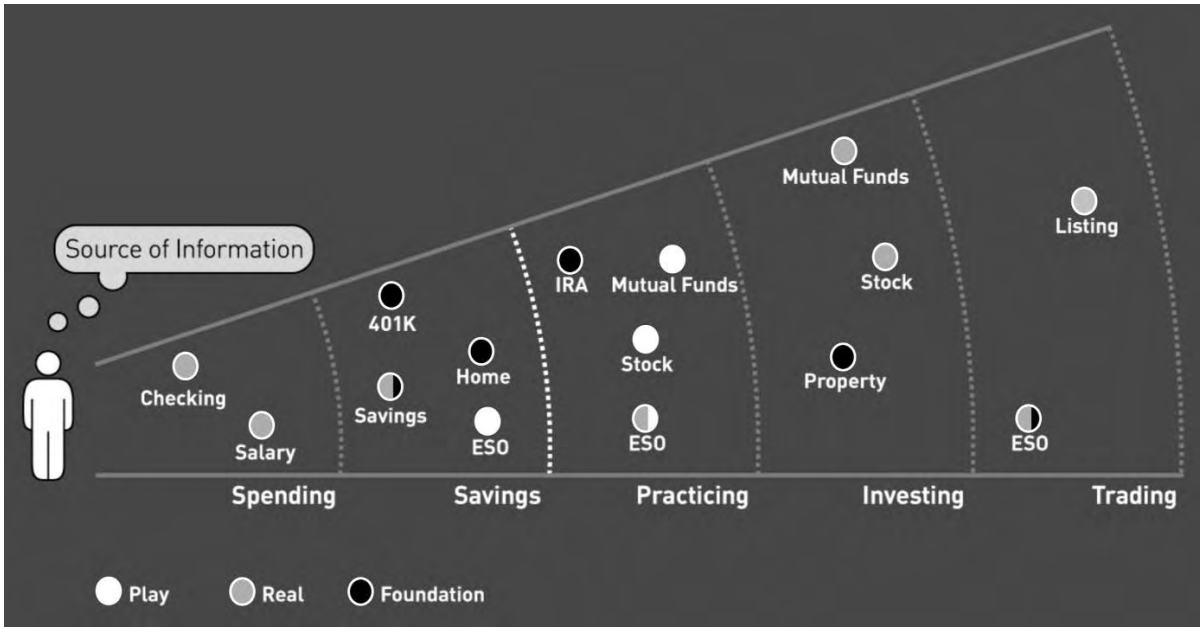


FIGURE 50.2. Financial development zones model.

the varied meanings of "money." This particular model highlighted the role of "practice" in developing the confidence and knowledge to become engaged in the investment process, and the iterative/recurrent nature of the process, as people learned to deal with new financial instruments and domains (e.g., securities, bonds, options, etc.). Moreover, it illustrated the distinctions that people make between "real," "play," and "foundational" money and the relationship between these categories, investment behavior, and financial development. To oversimplify a bit, people are more fully engaged and active in the investment process when they view the assets/investments as "real" (e.g., money that is used to address their current and emerging needs, pay bills, etc.) rather than as "play" (e.g., stock options that are perceived as intangible and somewhat imaginary) or "foundational" (e.g., savings for the future that are left "untouched"). As people have an opportunity to "practice" and develop their knowledge, they may move from construing a particular financial instrument or activity as "play" to "real." These notions suggested that web applications in this domain should not be focused on simply providing a wealth of financial information or a plethora of tools. Instead, these patterns helped to foster the generation of numerous ideas about ways to engage people in playful learning in the financial domain, with the aim of facilitating the financial development process.

PROCESS MODELS

Process models attempt to represent how a dynamic experience "works" and/or unfolds over time. They can range in focus from relatively circumscribed task-flow models that outline how an individual completes a specific task, to broader characterizations of more holistic change processes (e.g., healthcare behavior change, technology adoption, etc.). For example, a health services company aimed to develop an "electronic medical record system" (combining client server applications with web based "portals"). This system would, among other things, increase the efficiency and effectiveness of their medical practice, enable patients to view their health records online, and ultimately empower patients and foster a proactive approach to wellness and healthcare (both by clinicians and patients).

At the outset of the engagement, the health services company had generated a rather long requirements list (several hundred features and functions) and a particular view of the structure and function of the web components of the system. It was clear that the budget for this initiative was not sufficient to build a system that met all of the initial "requirements." Perhaps more importantly, it was unclear which components would ultimately add the most value for the various stakeholders (clinicians, patients, the business owners, etc.). Ethnographic research examining the experiences of and relationships between clinicians and patients in context (in clinic settings and in homes) provided the means of prioritizing and evaluating potential features, functions, and design concepts.

Experience models of varying levels of complexity regarding the health management process were developed. For example, one of the simpler models (see Fig. 50.3) described how individuals, in the process of adopting an active/proactive stance in relation to health issues, move through varying "stages of readiness." A more comprehensive, integrative model highlighted the ways in which various factors interact in influencing a person to take action in addressing a health issue and mapped the role of various healthcare related activities (e.g., monitoring, motivating, learning, sharing, building rapport) in various stages of readiness. The combination of these models enabled the team to identify the most important opportunities for facilitating progression towards a proactive orientation to health, and provided guidance in identifying ways to provide messages and experiences tailored to a person's stage and readiness.

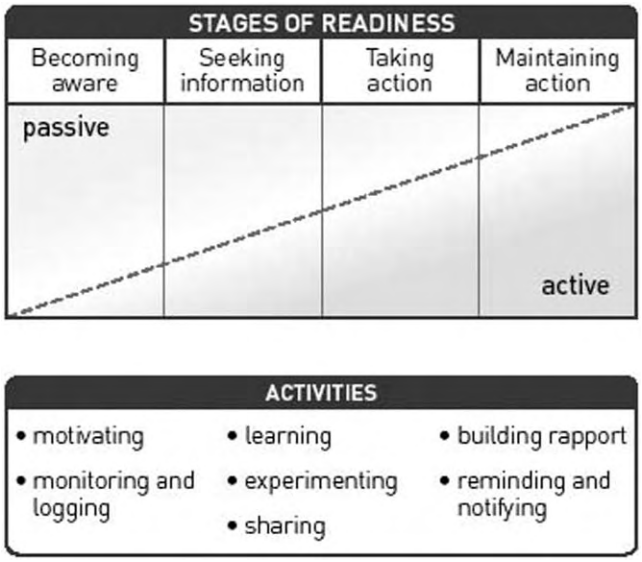


FIGURE 50.3. Stages of readiness model.

PERSONAS

One of the primary challenges in developing interactive systems is to design them so that they meet the needs of varying users, who may play different roles, engage in varied tasks, have different motivations and strategies, and so forth. Profiles or personas are abstract representations of the users of a solution (Pruitt & Grudin, 2003) which may be informed through ethnographic studies. Personas can help development teams understand and anticipate how certain types of people may experience and interact with technology solutions. For example, Fig. 50.4 shows a simple persona developed to guide the design of interactive tools promoting the adoption of various financial and health benefit programs in a large enterprise. Note that the persona focuses characteristics (attitudes, life stages, scenarios, etc.) that are most relevant to the person's experiences in managing financial and health-related concerns.

The value of personas can be enhanced by making them visible and dynamically present for design and development teams (e.g., posters displayed in project rooms, multimedia representations that are reviewed with development teams, role-playing scenarios and walkthroughs based on profiles, etc.). Rich and

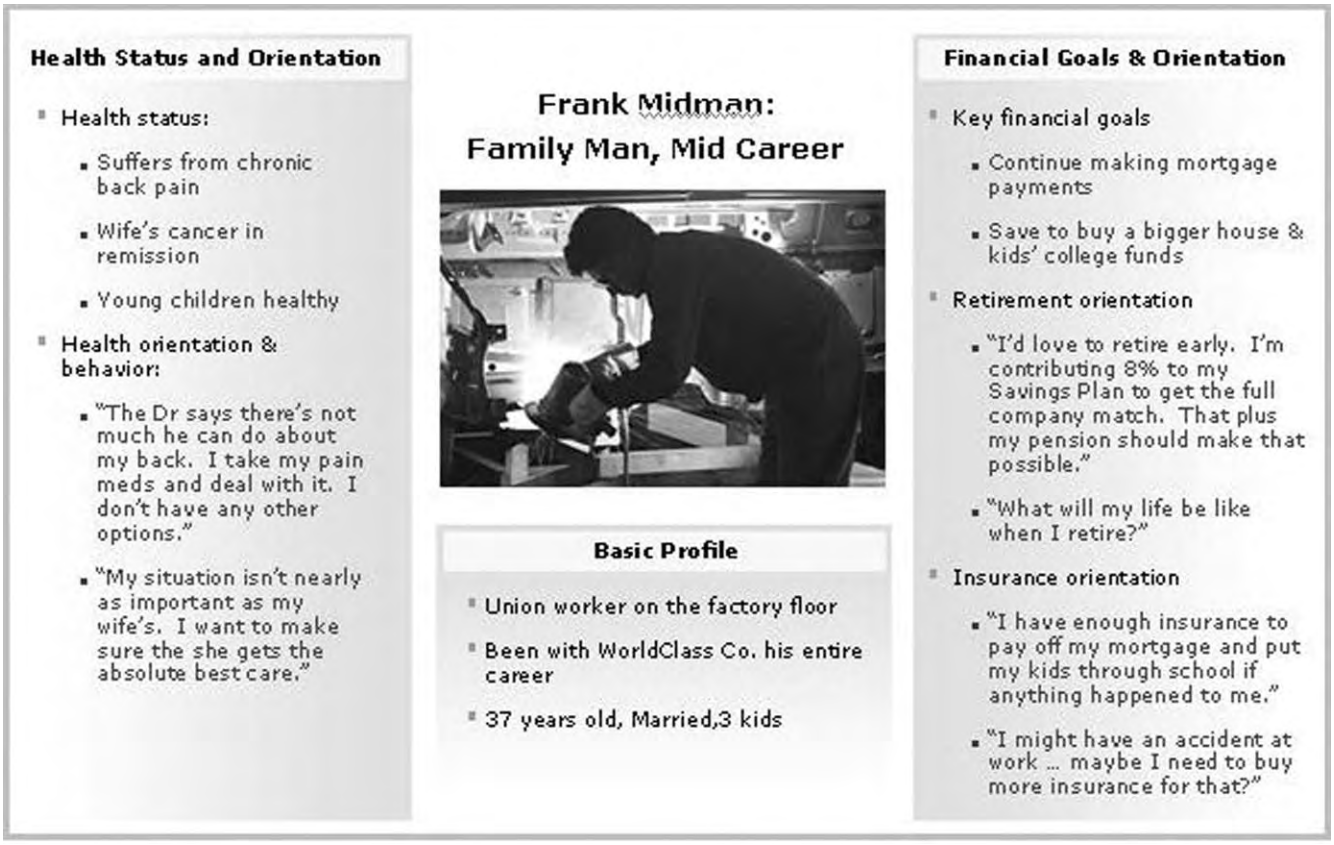


FIGURE 50.4. Financial and health benefit program design persona.

dynamic representations of essential characteristics of individuals can serve as a common frame of reference for communication and a tangible reminder to development teams regarding the people for whom they are designing the system. Moreover, personas can be used systematically in a range of ways to help teams make design decisions. For example, Pruitt and Grudin (2003) described specific techniques they have used to systematically apply personas to aid in feature prioritization decisions.

SCENARIOS

Scenarios are another way ethnographic research findings can be portrayed (Carroll, 2000; Nardi, 1992; Sonderegge, Manning, Charron, & Roshan, 2000; Rosson & Carroll Chapter X). Scenarios illustrate experiences and actions unfold in specific contexts or situations (Fig. 50.5) and can be documented in various forms ranging from narratives to annotated visual flow diagrams. They may highlight interactions (with computer systems, people, business entities, etc.), decisions processes, activity sequences, influencing factors, and so forth. They also may illustrate the different ways in which varied groups or types of people experience and navigate through similar situations. Analysis of scenarios can foster the identification of areas of difficulty ("pain points") and experiential gaps (or opportunities), that may be addressed or enhanced through various design solutions. When integrated with personas, they can illustrate how different target audiences navigate through the same situation, which in turn can suggest ways in which solutions can and should be adapted for varying target audiences.

MOCK-UPS AND PROTOTYPES

Representational artifacts, be they paper prototypes, mock-ups, or working prototypes, can play an important mediating role in connecting use requirements and design possibilities. When informed by studies of practice, these design representations respecify practices and activities in ways that are recognizable to practitioners. The prototypes go beyond simple demonstrations of functionality to incorporate materials from the participants' site, embody envisioned new technological possibilities, convey design ideas in relation to existing practices, and reveal requirements for new practices. Prototyping practices as such recover and invent use requirements and technological possibilities that make sense each in relation to the other (Suchman, Blomberg, & Trigg, 2002). In addition these representational artifacts facilitate the communication of what has been learned about technologies-in-use to the larger research and technology-development communities.

In an ethnographic study of engineering practice at a state Department of Highways, design prototypes critically deepened

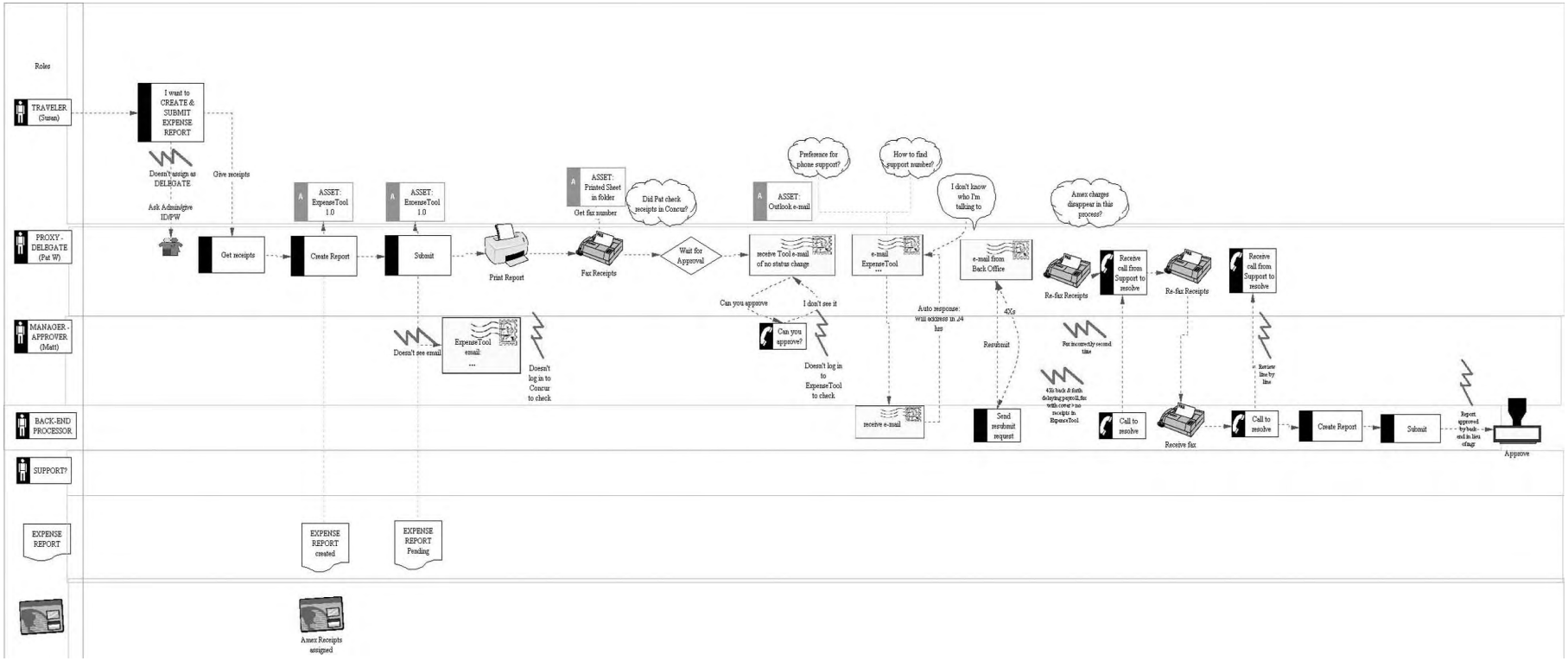


FIGURE 50.5. Scenario flow model.

the researchers understanding of the requirements of the work of document filing and retrieval (the focus of the study). At each step, from early design discussions with practitioners, to the creation of paper "mockups" of possible interfaces to the online project files, and finally to installing a running system at the worksite, the researchers became more aware of the work's exigencies. For example, in recognition of some of the difficulties that engineers experienced with their filing system various alternative document-coding strategies that augmented the existing filing system were designed. Through successive rounds, in which engineers were asked to code documents using mocked-up coding forms (both paper-based and online) the researchers understanding of the requirements of the work deepened. Eventually, the search and browsing interfaces evolved to be more finely tuned to the requirements of the engineers' work (e.g., Trigg, Blomberg, & Suchman, 1999).

CAVEAT REGARDING REPRESENTATIONS AND MODELS

Although representations and models are valuable tools for connecting ethnographic understanding and design, they can also have negative effects. Although grounded in observations and other forms of ethnographic inquiry, models are always a selective interpretation and construction of experience. Thus, while representations and models can focus attention on and illuminate important aspects of experience, they can also become reified stereotypes and constraints that inhibit design possibilities. Ongoing inquiry, a critical perspective, and a willingness to evolve the representations in the face of new learning are essential to maintain the viability and value of models for design.

RELATION TO OTHER QUALITATIVE APPROACHES AND PERSPECTIVES

The ethnographic approach has strong connections to and affinities with other approaches that have contributed to the development of the field of human-computer interaction, namely distributed cognition, activity theory, ethnomethodology, and participatory design. There is not space here to go into depth on any of these approaches. Our aim is simply to highlight relations between these approaches and ethnography, and provide a way to distinguish between them.

Distributed cognition (sometimes referred to as social or situated cognition) was first introduced to the HCI community by Lave (1988) and Hutchins (1995). Distributed cognition located cognition in social and material processes. When it was introduced, it challenged the dominant paradigm within HCI, that cognition primarily involved the psychological and mental processes of individuals. The connection between distributed cognition and ethnography is not only in the insistence that our understanding of human activity be located outside individual mental processes, in human interaction, but also in the conviction that to gain an understanding of human activity, ethnographic, field-based methodologies are required.

Activity theory also shares with ethnography a commitment to field-based research methodologies. In addition, there is the shared view that behavior (activity) should be a primary focus of investigation and theorizing, and a recognition that objects (artifacts) are key components in descriptive and explanatory accounts of human experience (e.g., Engeström, 2000; Nardi, 1996).

Ethnomethodology is often used interchangeably with ethnography in HCI literature. This is not only because the terms are etymologically similar, but also because many of the social scientists contributing to the field of HCI have adopted an ethnomethodological approach (e.g., Bentley et al., 1992; Button & Harper, 1996; Crabtree, 2000; Hughes et al., 1993, 1994, 1995) with its focus is on locally and interactionally produced accountable phenomena. Ethnomethodology's particular set of commitments (e.g., Heritage, 1984) are not shared however by everyone working within the ethnographic paradigm.

Participatory design does not have its roots in qualitative social science research, but instead developed as a political and social movement, and as a design approach committed to directly involving end users in the design of new technologies (See Muller, this volume; also Schuler & Namioka, 1993; Kensing & Blomberg, 1988). Within the HCI context, participatory design has shed some of its political and social-action underpinnings, and often is viewed primarily as a set of methods and techniques for involving users in design. Its connection to ethnography is in the commitment to involve study participants in the research, and in the value placed on participants' knowledge of their own practices. Also in recent years, those working in the field of participatory design have incorporated ethnographic techniques (e.g., Crabtree, 1998; Kensing, Simonsen, & Bødker, 1999) as a way of jointly constructing with participants knowledge of local practices.

ETHNOGRAPHY IN ACTION

Case Study 1: Designing a Program and Website to Change Healthcare Behaviors

A large global company, providing health insurance coverage to over 60,000 of its employees in the United States, developed a multifaceted program to reduce its healthcare costs and optimize the health and productivity of its workforce. The major goals were to provide reliable healthcare information and to promote better healthcare decisions. The program provided a number of online and offline resources for employees (e.g., a 24-hour medical hotline, a research team that would provide gather and summarize treatment outcome research findings for severe medical conditions, online access to a leading edge medical information/content website, etc.). The company initially promoted the program through a series of face-to-face workshops designed to convey the limitations of standard medical practice, encourage a consumer-oriented approach to healthcare, and make people aware of resources provided by the company.

After the initial launch, the team became concerned that the health program resources, including the website were being underutilized by employees, limiting the potential impact and value for both employees and the company. In this context, the

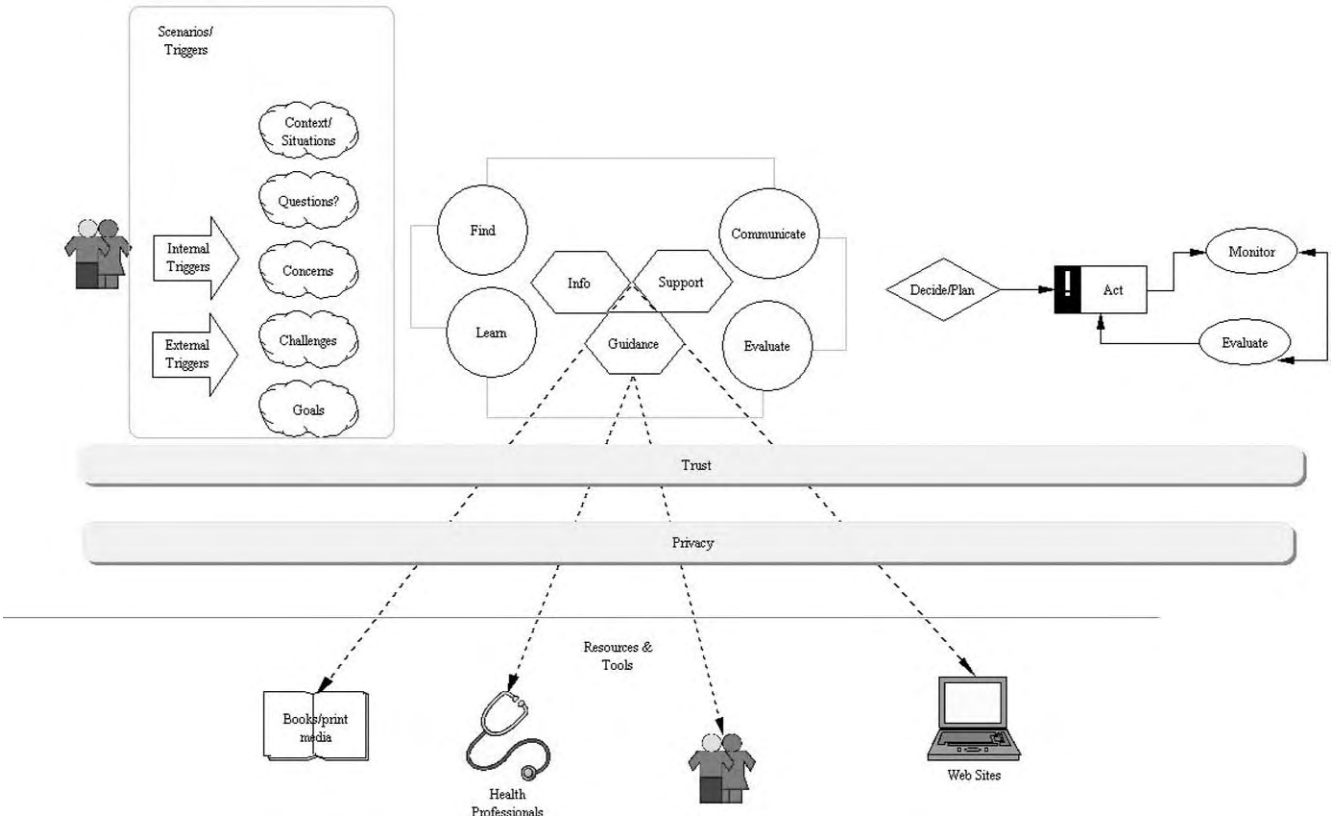


FIGURE 50.6. Healthcare decision-making model.

team initiated a study to evaluate the current program and website as well as well as to establish clear user models and strategic frameworks to guide website/program redesign efforts. To meet the project objectives, the research team conducted ethnographic inquiries combined with scenario-based exercises. In order to extend the participant sample as well as to deal with practical constraints (very limited time and resources), the team conducted some of the interviews and assessments remotely, via telephone and web-conferencing tools. Ethnographic inquiries focused on understanding the varied ways that people managed their healthcare (and/or the healthcare of family members), including their overall orientations to health and wellness, relationships and interactions with healthcare providers (and other family members), and their healthcare-decision-making processes. The latter included understanding the online and offline resources and tools that people used and the major healthcare scenarios they addressed. After exploring and profiling participants' healthcare experiences, they were asked to work through an actual healthcare decision scenario, while being invited to engage with the program resources and website.

Based on these inquiries, the research team developed a number of experience models including: a set of personas highlighting key variations in healthcare orientation and behavior that the program/website design team would have to accommodate; a simple typology of health-related scenarios (e.g., managing severe and chronic medical conditions, dealing with common everyday healthcare issues, and "wellness"/risk reduction); scenario flow models (Fig. 50.5) documenting how varied types of people made decisions (Fig. 50.6) and used a range of resources to address key health scenarios.

These models along with other resources generated numerous insights about limitations of the current website and program, opportunities for program/website enhancement, and design recommendations. For example, user profiles and scenario models showed how the program was fragmented and did not effectively align with people's key health scenarios, forcing an individual to painfully sift through resource information and descriptions to figure out which resources might be most relevant and useful in a specific scenario. In addition, the program and the website did not adequately address "wellness"/risk reduction scenarios which represented a significant concern for almost all employee segments and presented an important opportunity for the company to promote a proactive and preventative approach to healthcare.

In order to connect the user insights with the program/website design, the team articulated a number of design principles and a specific scenario-based design framework (Fig. 50.7). This framework highlighted the value of organizing the website (and other program elements) based on key healthcare scenarios, aligning and prioritizing resources and inviting specific modes of action that were most important in each scenario, and enabling relevant "cross-scenario" awareness

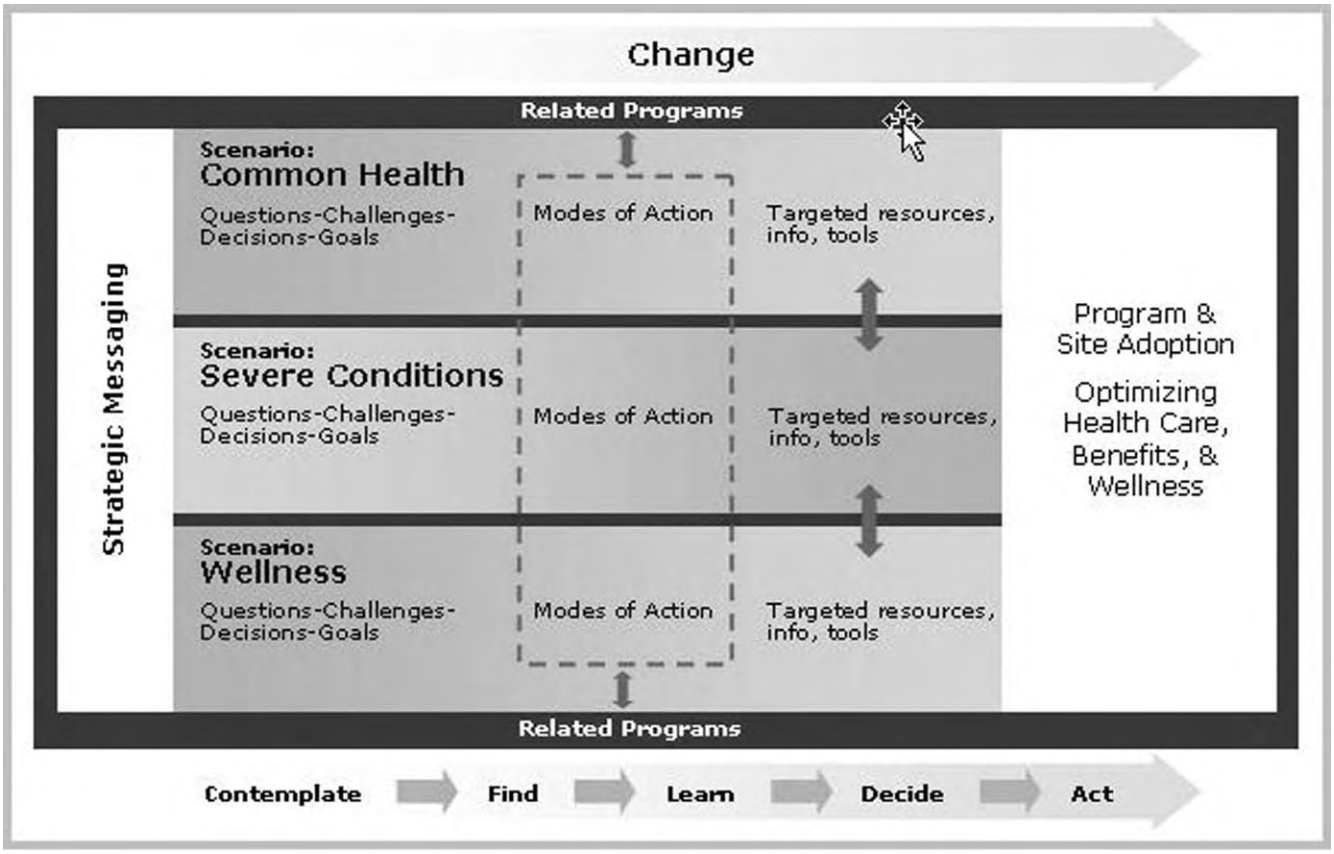


FIGURE 50.7. Scenario-based design framework.

and behavior that would provide value to users and support program objectives (e.g., a person who came to the website to learn about current research findings on the effectiveness of an experimental treatment for diabetes, might also be invited to explore the value of changes in diet or exercise to manage diabetes, etc.). In addition, the framework highlighted the importance of embedding strategic messages regarding healthcare (e.g., importance of evidence-based medicine, proactively taking charge of one's health and healthcare, etc.) and implicit invitations to change healthcare behaviors throughout the site design.

The ethnographic research led the team to rethink a number of major assumptions, which in turn led to redesign of program strategies, resources, and the website. From a program perspective, the research highlighted the fact that the vast majority of employees had already adopted many consumer attitudes and behaviors and were leveraging a number of trusted health resources (in contrast to initial assumptions of limited "consumerism"). This led the team to reconsider the positioning of specific program resources, shift strategic messaging, and generate novel program strategies including behavioral "rewards" programs that supported proactive and preventative behaviors. The initial research inquiry also led the team to implement a continuous assessment program to continue to monitor program impact and changes in employee experiences and behaviors.

Case Study 2: Department of Highways

The headquarters of a state Department of Highways was the site for a collaborative research and design effort with engineers charged with the design of a bridge, scheduled for completion by the year 2002.[18] The project aimed to design an electronic document-management system that was informed by an understanding of the everyday requirements of engineering work at the Department of Highways. The project began with onsite interviews and observations of engineering practice, with a focus on the document-related work practices.

Based on an initial understanding of the document-management requirements of the work, as part of the design process several alternate paper-based document-coding forms were designed. After several iterations, a coding form was settled upon that was then incorporated into the electronic document management system, both as a form to be scanned into a document database and as a model for an online coding form. The

[18]For more on the project with the Department of Highways see Suchman (1999, 2000).

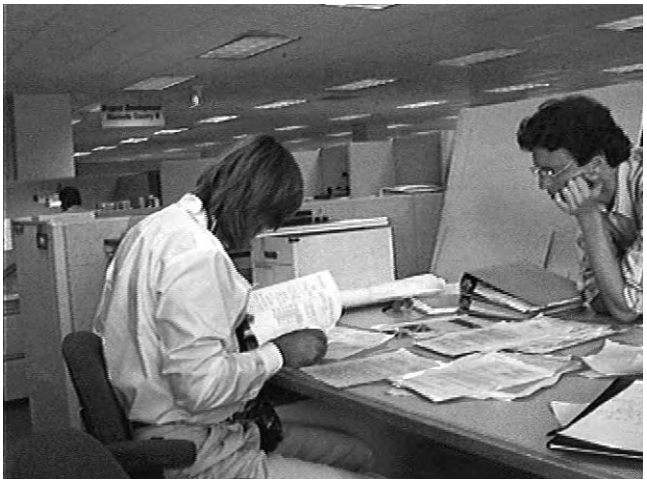


FIGURE 50.8. Engineer using mock-up of coding form to code documents.

evolution of the coding form was informed by the prompted use of the form by engineers at the Department of Highways (Fig. 50.8).

One of the key insights that came from the ethnographic study was the need to design continuing connections between the digital and physical document worlds. This included locating familiar ways of organizing documents in the new electronic system, and taking advantage of visual memory in document search and browsing by displaying page images of the documents and not just the text (Fig. 50.9).

The insights gained from the ethnographic study also pointed to challenges that would face engineering teams adopting the new system. First, because members of project teams would no longer be the sole interface to the documents in the project files, team members would need to consider who might view the documents and for what purposes before deciding to add a document to the database. This was not necessary when the project files were paper-based because the physical location of the documents, in the engineering team's work area, restricted access. Electronic access now meant that users of the system could be located anywhere within the Department of Highways, making explicit access controls necessary. In addition, it would be crucial that an ongoing relation between the paper and digital document renderings be maintained as engineers found it most useful to work with the printouts of large engineering documents. The online renderings were not particularly useful by themselves. The research and design team was able to anticipate these work practice issues, make the highways engineers aware of them, and suggest possible ways they could be addressed.

CONCLUSION

Ethnographic studies have become an important tool for designers and development teams designing new information and communication technologies. Today in academic, institutional, and corporate settings there is the realization that understanding the everyday realities of people living and working in a wide range of environments and engaged in a myriad of activities is essential for creating technologies and services that provide



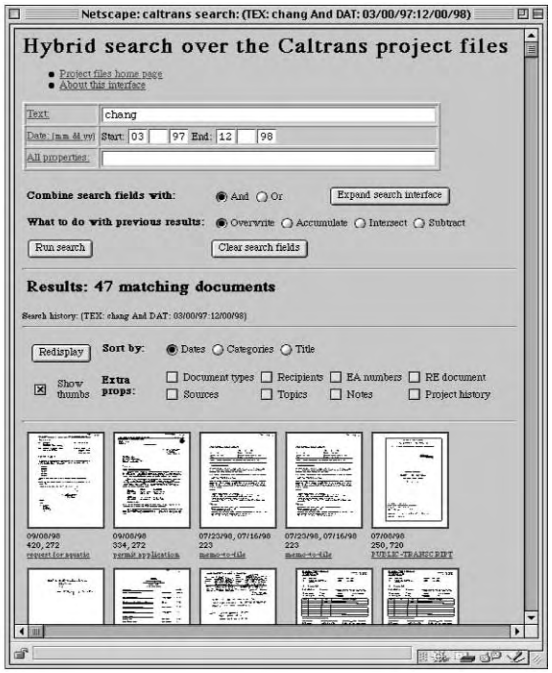


FIGURE 50.9. Components (document scanner, PC, coding forms, etc.) of the designed document management system and document search results page with thumbnails.

engaging and productive experiences for their users.[19] Emerging from recent research and practical experience is the recognition that representational tools (models, personas, scenarios, mock-ups and prototypes, etc.) and design-and-development practices (collaborative data analysis, video review sessions, etc.) are necessary for connecting ethnographic studies and technology design. Insights from ethnographic studies do not map directly onto design specifications or straightforwardly generate "user" requirements. Instead ethnographic studies must be connected and integrated with design agendas and practices. Those wishing to leverage the potential of ethnographic studies should not only understand what motivates the approach and is at its foundation (e.g., natural settings, holistic, descriptive, members' point of view), but also should recognize the importance of creating the conditions in which design can take advantage of ethnographic insights.

References

Anderson, K., & Lovejoy, T. (2005). *Proceedings of EPIC 2005*. Berkeley: University of California Press.

Agar, M. (1996). *The professional stranger* (2nd ed.). San Diego, CA: Academic Press.

Anderson, R. J. (1994). Representations and requirements: The value of ethnography in system design. *Human-Computer Interaction, 9*(2), 151–182.

Babbie, E. (1990). *Survey research methods* (2nd ed.). Belmont, CA: Wadsworth Publishing Company.

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., et al. (1992). Ethnographically-informed system design for air traffic control. In *Proceedings of Computer Supported Cooperative Work* (pp. 123–129). New York: ACM Press.

Bernard, H. R. (1995). *Research methods in anthropology: qualitative and quantitative approaches* (2nd ed.). London: Altamira Press.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems*. San Francisco, CA: Morgan Kaufmann Publishers.

Blomberg, J. (1987). Social interaction and office communication: Effects on user's evaluation of new technologies. In R. Kraut (Ed.), *Technology and the transformation of white collar work* (pp. 195–210). Hillsdale, NJ: Lawrence Erlbaum Associates.

Blomberg, J. (1988). The variable impact of computer technologies on the organization of work activities. In I. Greif (Ed.), *Computer-supported cooperative work: A book of readings* (pp. 771–782). San Mateo, CA: Morgan Kaufmann.

Blomberg, J. (1995). Ethnography: Aligning field studies of work and system design. In A.F. Monk & N. Gilbert (Eds.), *Perspectives on HCI: Diverse approaches* (pp. 175–197). London: Academic Press LTD.

Blomberg, J., Giacomi, J., Mosher, A., & Swenton-Wall, P. (1991). Ethnographic field methods and their relation to design. In D. Schuler & A. Namioka (Eds.), *Participatory design: Perspectives on systems design* (pp. 123–155). Hillsdale, NJ: Lawrence Erlbaum Associates.

Blomberg, J., Suchman, L., & Trigg, R. (1996). Reflections on a work-oriented design project. *Human-Computer Interaction, 11*(3), 237–265.

Blomberg, J., Suchman, L., & Trigg, R. (1997). Back to work: Renewing old agendas for cooperative design. In M. Kyng & L. Mathiassen (Eds.), *Computers and design in context* (pp. 267–287). Cambridge, MA: MIT Press.

Blomberg, J., & Trigg, R. (2000). Co-constructing the relevance of work practice for CSCW Design: A case study of translation and mediation. *Occasional Papers from the Work Practice Laboratory*, Blekinge Institute of Technology, *1*(2), 1–23.

Briggs, C. (1983). *Learning how to ask: A sociolinguistic appraisal of the role of the interview in social science research*. Cambridge, U. K.: Cambridge University Press.

Brun-Cotton, F., & Wall, P. (1995). Using video to re-present the user. *Communications of the ACM, 38*(5), 61–71.

Button, G., & Harper, R. (1996). The relevance of 'work-practice' for design. *Computer-Supported Cooperative Work, 5*, 263–280.

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.

Clifford, J. (1988). *The predicament of culture: Twentieth-century ethnography, literature, and art.* Cambridge, MA: Harvard University Press.

Clifford, J., & Marcus, G. (Eds.). (1986). *Writing culture: The poetics and politics of ethnography*. Berkeley: University of California Press.

Cohen, K. (2005). Who we talk about when we talk about users. *Proceedings of EPIC* (pp. 9–30). Arlington, VA: American Anthropological Association

Comaroff, J., & Comaroff, J. (1992). *Studies in the Ethnographic Imagination.* Boulder, CO: Westview Press.

Corral-Verduga, V. (1997). Dual 'realities' of conservation behavior: Self reports vs. observations of re-use and recycling behavior. *Journal of Environmental Psychology, 17*, 135–145.

Crabtree, A. (1998). Ethnography in participatory design. In *Proceedings of the Participatory Design Conference* (pp. 93–105). Seattle, WA: Palo Alto, CA: CPSR.

Crabtree, A. (2000). Ethnomethodologically informed ethnography and information system design. *Journal of the American Society for Information Science, 51*(7), 666–682.

Cross, R., & Parker, A. (2004). *The hidden power of social networks: Understanding how work really gets done in organizations*. Boston, MA: Harvard Business School.

D'Andrade, R. G. (1995). *The development of cognitive anthropology*. Cambridge, UK: Cambridge University Press.

Engeström, Y. (2000). From individual action to collective activity and back: Developmental work research as an interventionist methodology. In P. Luff, J. Hindmarsh, & C. Heath (Eds.), *Workplace studies: Recovering work practice and informing system design* (pp. 150–166). Cambridge, UK: Cambridge University Press.

Fluehr-Lobban, C. (Ed.). (1991). *Ethics and the profession of anthropology: Dialogue for a new era*. Philadelphia: University of Pennsylvania Press.

Green, E. C. (2001). Can qualitative research produce reliable quantitative findings? *Field Methods, 13*(1), 1–19.

Green, S., Harvey, P., & Knox, H. (2005). Scales of place and networks: An ethnography of the imperative to connect through information and communication technologies. *Human Organization, 46*(5), 805–826.

Grief, I. (Ed.). (1988). *Computer-supported cooperative work: A book of readings*. San Mateo, CA: Morgan Kaufmann.

[19]For a discussion of the relation between ethnography and design see also Anderson, (1994), Grudin and Grintner (1995), Rogers and Belloti (1997), and Shapiro (1994).

Grudin, J., & Grintner, R. E. (1995). Ethnography and design. *Computer Supported Cooperative Work, 3*(1), 55–59.

Gubrium, J. F., & Holstein, J. A. (Eds.). (2002). *Handbook of interview research: Context and method*. Thousand Oaks, CA: Sage Publication.

Guest, G. (2000). Using Guttman scaling to rank wealth: integrating quantitative and qualitative data. *Field Methods, 12*(4), 346–357.

Harding, S. (1986). *The science question in feminism*. Ithaca, NY: Cornell University Press.

Heritage, J. (1984). *Garfinkel and ethnomethodology.* Cambridge, MA: Polity Press.

Horst, H., & Miller, D. (2005). From kinship to link-up: Cell phones and social networking in Jamaica. *Human Organization, 46*(5), 755–778.

Hughes, J. A., Randall, D., & Shapiro, D. (1993). From ethnographic record to system design: Some experiences from the field. *Computer Supported Cooperative Work, 1*(3), 123–147.

Hughes, J. A., Rodden, T., & Anderson, H. (1995). The role of ethnography in interactive system design. *ACM Interactions, 2*(2), 56–65.

Hutchins, E. (1995). *Cognition in the wild.* Cambridge, MA MIT Press.

Johnson, J. C., & Griffith, D. C. (1998). Visual data: collection, analysis, and representation. In V. DeMunck & E. Sobo (Eds.), *Using methods in the field: a practical introduction and casebook* (pp. 211–228). Walnut Creek, CA: Altamira.

Johnson, J. C., Ironsmith, M., Whitcher, A. L., Poteat, G. M., & Snow, C. (1997). The development of social networks in preschool children. *Early Education and Development, 8*(4), 389–406.

Jordan, B., & Suchman, L. (1990). Interactional troubles in face-to-face survey interviews. *Journal of the American Statistical Association, 85*(409), 232–253.

Kantner, L. (2001). Assessing website usability from server log files. In R. Branaghan (Ed.), *Design by people, for people: Essays on usability* (pp. 245–262). Usability Professionals Association. Chicago, IL.

Karasti, H. (2001). Bridging work practice and system design—integrating systemic analysis, appreciative intervention, and practitioner participation. *Computer Supported Cooperative Work—An International Journal, 10*(2), 167–198.

Kensing, F., & Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work, 7*(3–4), 163–165.

Kensing, F., Simonsen, J., & Bødker, K. (1999). MUST—a method for participatory design, *Human-Computer Interaction, 13*(2), 167–198. Mahwah, NJ: Lawrence Erlbaum.

Kilduff, M., & Tsai, W. (2003). *Social networks and organizations*. London: Sage Publication.

Koskinen, I., Battarbee, K., & Mattelmäki, T. (2005). *Emphathic design: User experience in product design*. Helsinki: IT Press.

Lave, J. (1988). *Cognition and practice.* Cambridge, U.K.: Cambridge University Press.

Latour, B. (1987). *Science in action: How to follow scientists and engineers through society*. Cambridge, MA: Harvard University Press.

Latour, B., & Woolgar, S. (1986). *Laboratory life: The construction of scientific facts*. Princeton, NJ: Princeton University Press.

Leonard, D., & Rayport, J. F. (1997). Sparking innovation through empathic design. *Harvard Business Review, 75*(6), 102–113.

Marcus, G., & Fischer, M. (1986). *Anthropology as cultural critique: An experimental moment in the human sciences*. Chicago: University of Chicago Press.

Masten, D., & Plowman, T. (2003). Digital ethnography: The next wave in understanding the consumer experience. *Design Management Journal*, *14*(2), 75–84.

Moore, R. J. (2004). Managing troubles in answering survey questions: Respondents' uses of projective reporting. *Social Psychology Quarterly,* 67(1), 50–69.

Nader, L. (1974). Up the anthropologist—perspectives gained from studying up. In D. Hymes (Ed.), *Reinventing anthropology* (pp. 284–311). New York: Vintage.

Nardi, B. (1992). The use of scenarios in design. *SIGCHI Bulletin, 24*(4), 13–14.

Nardi, B. (1996). *Context and consciousness: Activity theory and human-computer interaction.* Cambridge, MA: MIT Press.

Nardi, B., & Miller, J. (1990). An ethnographic study of distributed problem solving in spreadsheet development. In *Proceedings of Computer Supported Cooperative Work* (pp. 197–208). New York, NY: ACM Press.

Newman, S. E. (1998). Here, there, and nowhere at All: Distribution, negotiation, and virtuality in postmodern engineering and ethnography. *Knowledge and Society, 11*, 235–267.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of ACM INTERCHI'93 Conference* (pp. 206–213). Amsterdam, The Netherlands: ACM Press.

Perkins, R. (2001). Remote usability evaluation over the Internet. In R. Branaghan (Ed.), *Design by people, for people: Essays on usability* (pp. 153–162). Usability Professionals Association. Bloomingdale, IL

Pickering, A. (Ed.). (1980). *Science as practice and culture*. Chicago: University of Chicago Press.

Pine, J., & Gilmore, J. (1999). *The experience economy: Work is theater and every business a stage*. Cambridge, MA: Harvard Business School Press.

Pruitt, J., & Grudin, J. (2003). Personas: practice and theory. In *Proceedings of Designing for User Experience* (pp. 1–15). New York: ACM Press.

Rathje, W. L., & Cullen Murphy, C. (1991). *Rubbish! The archaeology of garbage*. New York: HarperCollins.

Rich, M., Lamola, S., Amory, C., & Schneider, L. (2000). Asthma in life context: video intervention/prevention assessment (VIA). *Pediatrics, 105*(3), 469–477.

Rheingold, H. (2000). *Virtual community: Homesteading on the electronic frontier*. Cambridge, MA: MIT Press.

Robinson, R. E. (1994). The origin of cool things. In *Proceedings of the American Center for Design Conference on Design that packs a wallop: Understanding the power of strategic design.* (pp. 5–10). New York, NY American Center for Design.

Robinson, R. E., & Hackett, J. P. (1997). Creating the conditions of creativity. *Design Management Journal, 8*(4), 10–16.

Rodden, T., & Anderson, H. (1994). Moving out from the control room: Ethnography in system design. In R. Furuta & C. Neuwirth (Eds.), *Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 429–439). New York: ACM Press.

Rogers, Y., & Bellotti, V. (1997). How can ethnography help? *Interactions, 4*, 58–63.

Romney, A. K., Batchelder, W. H., & Weller, S. C. (1986). Culture as consensus: A theory of culture and informant accuracy. *American Anthropologist, 88*, 313–338.

Sachs, P. (1995) Transforming work: Collaboration, learning, and design. *Communications of the ACM, 38*(9), 36–44.

Said, E. (1978). *Orientalism*. New York: Pantheon.

Salant, P., & Dillman, D. A. (1994). *How to conduct your own survey*. New York: Wiley and Sons, Inc.

Schmidt, K., & Bannon, L. (1992). Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work, 1*(1–2), 7–40.

Scott, J. P. (2000). *Social network analysis: A handbook* (2nd ed.). London: Sage Publication.

Shapiro, D. (1994). The limits of ethnography: Combining social sciences for CSCW. In *Proceeding of Computer Supported Cooperative Work* (pp. 417–428). New York: ACM Press.

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory design: Principles and practices.* Hilldale, NJ: Lawrence Erlbaum Associates.

Smith, D. (1987). *The everyday world as problematic: A feminist sociology*. Boston: Northwestern University Press.

Sonderegger, P., Manning, H., Charron, C., & Roshan, S. (2000). Scenario design. In Forrester Report, December 2000.

Squires, S., & Byrne, B. (Eds.). 2002. *Creating breakthrough ideas: The collaboration of anthropologists and designers in the product development industry*. Westport, CT: Bergin & Garvey.

Suchman, L. (1983). Office procedures as practical action: Models of work and system design. *ACM Transactions on Office Information Systems, 1*(4), 320–328.

Suchman, L. (1999). Embodied practices of engineering work [Special issue]. *Mind, Culture and Activity, 7*(1–2), 4–18.

Suchman, L. (2000). Organizing alignment: A case of bridge-building. Organization, 7(2), 311 327.

Suchman, L., & Trigg, R. (1991). Understanding practice: Video as a medium for reflection and design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: cooperative design of computer systems* (pp. 65–89). Hillsdale, NJ: Lawrence Erlbaum Associates.

Suchman, L. Blomberg, J., & Trigg, R. (1999). Reconstructing technologies as social practice. American Scientist, 43(3), 392–408.

Suchman, L., Trigg, R. & Blomberg, J. (2000). Working artifacts: Ethrome thods of the prototype. British Journal of Sociology, 53(2), 163–179.

Thomke, S. (2003, April). R&D comes to services. *Harvard Business Review*, 71–79.

Trigg, R., Blomberg, J., & Suchman, L. (1999). Moving document collections online: The evolution of a shared repository. In *Proceeding of the European Conference on Computer-Supported Cooperative Work* (pp. 331–350). Copenhagen. Kluwer Academic Publishers, Norwell, MA.

Wasson, C. (2000). Ethnography in the field of design. *Human Organization, 59*(4), 377–388.

Whiting, B., & Whiting, J. (1970). Methods for observing and recording behavior. In R. Naroll & R. Cohen (Eds.), *Handbook of method in cultural anthropology* (pp. 282–315). New York: Columbia University Press.

Whyte, W. F. (1960). Interviewing in field research. In R. Adams & J. Preiss (Eds.), *Human organization research* (pp. 299–314). Homewood, IL: Dorsey.

Whyte, W. F. (1984). *Learning from the field: a guide from experience*. Newbury Park, CA: Sage.

Wolf, M. (1992). *A thrice-told tale: Feminism, postmodernism, and ethnographic responsibility.* Stanford, CA: Stanford University Press.

Yanagisako, S., & Delaney, C. (Eds.). (1995). *Naturalizing power: Essays in feminist cultural analysis*. New York: Routledge.

SECTION B

DESIGN AND DEVELOPMENT

·51·

PUTTING PERSONAS TO WORK: USING DATA-DRIVEN PERSONAS TO FOCUS PRODUCT PLANNING, DESIGN, AND DEVELOPMENT

Tamara Adlin
adlin, inc.

John Pruitt
Microsoft Corporation

Introduction **992**
It Is Hard to Be User-Centered **992**
The Word User Is Part of the Problem 992
Personas Move Us Beyond the Word "User" 993
Making Personas Practical: The Creation of the Persona Lifecycle Method **994**
The Persona Lifecycle **994**
Phase 1: Persona Family Planning 995
Organizational Introspection 995
Creating a Persona Core Team 995
Identifying Data Sources and Collecting Data 996
Phase 2: Persona Conception & Gestation 997
Step 1: Identify and Assimilate Assumptions 997
Get the rest of the assumptions out on the table 997
Step 2: Process Your Data 997
Identify key data points (factoids) in the data sources 998
Asssimilate the factoids 998
Step 3: Identify Subcategories of Users and Create Skeletons 999
Create skeletons 999
Step 4: Prioritize the Skeletons 1000
Step 5: Develop Selected Skeletons Into Personas 1001
Moving toward precision means moving away from accuracy 1002
Incorporate narrative and storytelling elements 1002
Illustrate your personas 1002
Step 6: Validate Your Personas 1003
Completed personas do not mark the end of user research 1003
Phase 3: Persona Birth & Maturation 1003
Persona Artifacts 1004
Agree on the specific goal of the artifact 1005
Agree on the audience, timing, and distribution method for the artifact 1005
Agree on the information elements that should (and should not) be included on the artifact 1005
Agree on the relative priorities of the information elements on the artifact 1006
Phase 4: Persona Adulthood 1006
Use Personas for Product Planning, Design, Evaluation, and Release 1007

Personas and product planning 1007
Personas and product design 1007
Personas and product design evaluation 1007
Personas and product release 1007
A Great Tool for Persona Adulthood: Using Design Maps to get From Personas to Product Designs 1008
What Are Design Maps? 1008
Which Processes Should I Design Map? 1008
Design map for the "big picture." 1008
Design maps for achieving major milestones . . . 1008
Design maps for critical details 1009
How to Create a Design Map 1009
The elements of a Map: steps, questions, comments, and design ideas 1009
Facilitating a Design Mapping session 1010
Use Design Maps to Create Wireframes 1010
The Benefits of Design Maps 1010
Phase 5: Persona Lifetime Achievement & Retirement 1012
Retirement 1013
Reusing Your Personas 1013
Reincarnating Personas 1013
Retiring Your Personas 1013
Moving on to the Next Persona Effort 1014
Summary 1014
Areas for Further Investigation 1015
References 1015

INTRODUCTION

Personas are a new and relatively popular technique used to help bring users into the forefront of people's minds during the difficult process of developing products and services. Yet, for many practitioners, the persona method remains largely ill defined and mysterious. In this chapter, we present a simple framework for approaching the technique—the *persona lifecycle*—which sheds light on how personas fit into a standard development cycle. We then provide a little detail on two critical aspects of the method: creating and using personas. Our goal is to enable the reader to quickly and easily get started with the technique.

IT IS HARD TO BE USER-CENTERED

In the best of all worlds, everyone working on a product would always be thinking of the needs of every person who will ever use the product. Real information about users would inform every decision and the resulting product would perfectly satisfy everyone who uses it. In practice, it is hard enough to get everyone working on a product to think about users at all (see Grudin, 1990, 1993, for a discussion of obstacles and constraints in product development organizations). To deliver on the promise and benefits of user-centered design, we have to find creative ways to inject accurate information about real users into the chaotic world of product development.

It's a rare product indeed that does everything you want it to do in the way you want to do it. Why? Because, despite the fact that building products based on what real people need and want seems obvious, putting users (or rather, information about users) truly at the center of the design and development process is extremely difficult. Why is it so hard to be user-centered?

First, being user-centered is just not natural. Our more natural tendency is to be self-centered—to design a product based on our own wants and needs (sometimes even if we are not actually a user of the product). Self-centered design is perhaps better than technology-centered design (another common inclination), but most of the time, the people on your product development team are not representative of the target audience for your product. Thus, as many in the user-centered design field have long evangelized (see, e.g., Nielsen, 1993), it is important to understand the user.

Second, users are complicated and varied. It takes great effort to understand their needs, desires, preferences, and behaviors. In addition, pleasing some of them does not necessarily result in pleasing others.

Third, those doing the user and market research to understand who the users are and how they vary (and others that are just more in touch your users—e.g., the sales team and the support team) are not typically the people who actually design and build the product. Those who collect the data try to communicate the information it contains by creating presentations or reports, which are helpful only if they are absorbed by the right people at the right times. If the important information about users is not available at the right time, or is difficult to understand or to remember, product teams forge ahead to design and build features they think the users would like, or more likely, what is easiest and least costly to build.

The Word U*ser* Is Part of the Problem

When user-centered design was a new idea, simply injecting the word *user* into a design and development process was powerful; it challenged the status quo. Unfortunately, incorporating the word *user* into everyday corporate discourse is not enough to foster effective user-centered design.

Everyone (we hope) assumes that they are building products with users in mind; in many organizations, anyone asked would probably answer, "Yes, I think about the user a lot." People who talk about the "user," however, are almost never asked to further define the term, and it is a sure bet that each person in the organization would describe the users in a different way. If every-

one in the organization doesn't have a clear and consistent understanding of *whom* they are building the product for, the product can fail. It is our contention that the word *user* cannot provide the clarity that is required.

> "User" is a catchall and ultimately a mean-nothing word. It reflects a technology-centric, rather than a people-centric, view of the web. To call someone a user is largely meaningless. The phrase "user-friendly" should never have had to be invented. It implies that technology is inherently hostile and that a new discipline—usability—had to be invented to make it friendlier. After all, we don't refer to cars as "driver-friendly."
>
> We don't refer to bicycles as "cyclist-friendly." We don't refer to chairs as "bum-friendly."
>
> —Gerry McGovern, Consultant, gerrymcgovern.com. (From "Don't call people users." April 1, 2002.)

Personas add the detail and specificity needed to provide product development teams the understanding needed create user-centered products.

Personas Move Us Beyond the Word "User"

Personas are fictitious, specific, concrete representations of target users. Personas put a face on the user; a memorable, engaging, and actionable image to serve as a design target. The term *personas* was originally adopted and popularized by Alan Cooper in his 1999 book, The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How To Restore The Sanity (see also Cooper & Reimann, 2003).

Personas were born out of a short tradition in the User-Centered Design (UCD) community toward user and system modeling and out of a somewhat longer tradition in marketing around market definition and customer representation. Perhaps the earliest comes from industrial designer Henry Dreyfuss wrote about "Joe and Josephine," the heroes of his book Designing for People (1955/2003), which were created to inform the physical design of objects and spaces. Geoffrey Moore, in his book Crossing the Chasm (1991), described the notion of "target customer characterizations"—images of customers meant to replace or supplement impersonal and abstract market segments (see for example, Sissors, 1966, or more recently, Weinstein, 1998). John Carroll has been a longstanding proponent of scenario-based design, which typically includes terse representations of users called actors or agents (see also, Jacobson, 1995; Jacobson, Christerson, Jonsson, & Övergaard, 1992, regarding "actors" in "use cases"). Carroll (1995, 2000a, 2000b) argued that scenarios help designers and analysts focus on assumptions about people and tasks.

These and many other representations of users or customers (e.g., see Constantine & Lockwood, 2001, 2002; Mello, 2003; Upshaw, 1995) are typically *not* well-rounded or complete descriptions, but instead are confined to a few key details and specific contexts. Moreover, they do not seem real; they are devoid of personality. Personas, on the other hand, seem like real people. As such, personas carry information about users to your product team in way that other representations cannot. Personas enable us to move beyond our habit of referring to "users" and instead allow us to find a better way to communicate about and focus on the real people we want to use our products. Personas humanize vast and disparate data sources, capitalize on our ability to remember details about individual people, and, in so doing, provide a usable and useful alternative to referring to "the user." In other words, personas create a concrete, focused, and stable definition of a specific audience.

While rigorous evaluation of these claims has not been provided to date, some case studies and anecdotal evidence support them to varying degrees (e.g., see Freed, 2004; Grudin & Pruitt, 2002; Hourihan, 2002; Levinson, 2003; see also the numerous sidebar stories in Pruitt & Adlin, 2006). Based on our own experience with the approach, we believe that when created with data and used in a thoughtfully during the product development process, personas:

- Make assumptions and knowledge about users explicit, creating a common language with which to meaningfully talk about users,
- Allow you to focus on and design for a small set of specific users (who aren't necessarily like you), helping you make better decisions, and
- Engender interest and empathy toward users.

The Genesis of Personas in Product Design: Cooper Takes "Play Acting" One Step Further

Kim Goodwin, Vice President and General Manager, Cooper

With the publication of The Inmates Are Running the Asylum in 1999, Alan Cooper introduced the world to personas as a practical interaction design tool; however, Alan and the folks at his leading design consultancy, Cooper, had already been using personas for years.

In 1983, Alan was working as a solo software inventor. While working on a project management program he called "Plan*It," Alan realized he needed to understand more about how project managers thought, so he interviewed a handful of people. A woman named Kathy, who seemed the most typical, was the basis for the first persona-like model in Alan's head. While waiting for his program to compile, Alan would play-act a project manager very much like Kathy, using the way she thought and worked to make decisions about the design of the application. Alan eventually sold Plan*It to Computer Associates, who sold it to the public as SuperProject. After that success, Alan went on to use this technique on other projects, including the visual programming language that became Visual Basic.

Later, when Alan ventured into consulting, he found that he could not just do what seemed right, because he first had to persuade other people. For this reason, on a 1995 project with Sagent Technologies, Alan created Chuck, Cynthia, and Rob—the first real, Goal-Directed personas. After some initial resistance, they worked as Alan had intended: they provided

a way to keep everyone focused on what users really wanted to do, rather than on all the things they might do.[1]

Since then, Cooper designers have refined and formalized the methods for researching, creating, and applying personas over the course of hundreds of projects. Many practitioners have begun using personas—some with excellent results, others not. The most common reason for failure? People miss the thing that makes personas so uniquely effective: they're based on a qualitative understanding of how real people behave and—equally important—why they behave that way.

Today's best personas are well researched, focused on behavior, and documented as a short story writer would describe a beloved character: with sympathy, respect, and just enough back story to help you understand what makes them tick.[2] Effective personas are based on the kind of information you can't get from demographics, survey data, or suppositions, but only from observing and interviewing individual people in their own environments. That qualitative, first-hand information is not only essential to design—it is essential to persuasion. If you cannot effectively persuade the programmers to build it, the executives to fund it, and the marketers and sales people to sell it, then the best design in the world is a failure. Personas are not just effective because they are accurate representations of human behavior. They are effective because they help both designers and stakeholders understand user needs at a gut level, which—in spite of all you hear about ROI—is where most business decisions are made.

MAKING PERSONAS PRACTICAL: THE CREATION OF THE PERSONA LIFECYCLE METHOD

If personas are such a good thing, why isn't everyone using them? Perhaps one answer is that creating and using personas is easier said than done. When we first started talking to persona practitioners in 2000, we noticed that many people in the UCD community could see the value of personas in their own work and to their organizations. Those that tried to create and use personas, however, were running into a somewhat consistent set of problems in their persona efforts. While the idea of creating a set of target users is reasonably straightforward, the actual process to create and use personas can be quite complex. We heard many of the same questions repeatedly:

- How do you decide whether personas are the right thing to do in your organization?
- How do you incorporate data into personas? What kinds of data work?
- How do you know if personas are worth the effort it takes to create them?
- How do you communicate personas once they are created?
- How do you use personas to design great products?
- What do you do with personas once a project is finished? Can you reuse personas?

As we continued our research over the following years, we discovered that many practitioners were having less-than-stellar experiences with personas. Those that were able to create data-driven personas were finding that, if they are not well communicated and managed, even well-crafted personas are easy for designers and developers to ignore. At worst, poorly executed persona efforts yield no increase in user-focus and leach time and resources from other UCD techniques and other methods that can improve product quality. We found four common reasons for the failure of persona efforts:

1. The effort was not accepted or supported by the leadership team,
2. The personas were not credible and not associated with methodological rigor and/or data,
3. The personas, and the techniques for using personas, were not well communicated,
4. The product design and development team did not understand how to use the personas.

Once we fully understood the questions and common causes of failure, we focused our attention on finding solutions based on the input and insights of dozens of persona practitioners. The persona lifecycle was the result.

THE PERSONA LIFECYCLE

The persona lifecycle is a metaphorical framework that breaks down persona creation and use into sequential phases that map on to the life stages of human reproduction and development. This framework is composed of five phases: (1) Family Planning, (2) Conception & Gestation, (3) Birth & Maturation, (4) Adulthood, and (5) Lifetime Achievement & retirement (see Fig. 51.1, below). The phases of the persona life cycle framework bring structure to the potentially complicated process of persona creation and highlight critical (yet often overlooked or ignored) aspects of persona use.

As the name indicates, the persona lifecycle is a cyclical, mostly serial process model. As the illustration in Fig. 51.1 shows, each stage builds on the next, culminating, but not ending at, Adulthood. Notice that the illustration also shows that the final stage, Lifetime Achievement & Retirement, is not immediately followed by the first stage. This is because different persona efforts culminate and restart in different ways; personas can be reused, reincarnated, or retired, depending on the project.

[1]For more on the origin of personas, see Alan Cooper's article on www.cooper.com.

[2]For more on creating better personas, see Kim Goodwin's "Perfecting Your Personas" and "Getting from Research to Personas" on www.cooper.com.

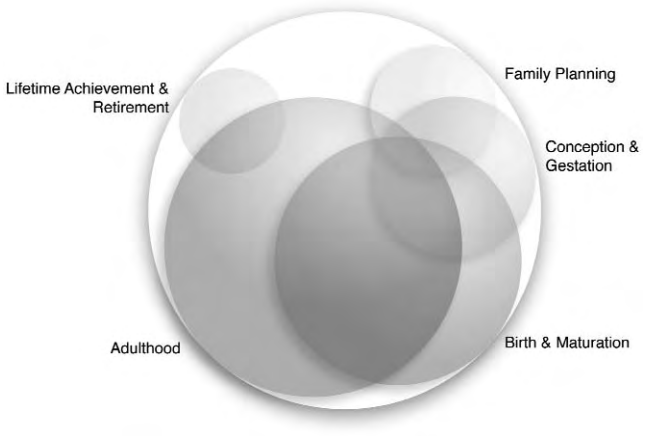


FIGURE 51.1. The five phases of the persona lifecycle. This diagram is designed to show both the order of the phases (from Family Planning through Conception & Gestation, Birth & Maturation, Adulthood, and finally, Lifetime Achievement & Retirement) and the relative amount of effort related to each phase.

If your company has already adopted some user-centered design methods, you should find that the phases of the persona lifecycle augment your existing process and help you get involved earlier in the product development cycle. Keep in mind that the persona lifecycle is not meant to replace other user-centered design tools and is not a complete user-centered product design method on its own. Rather, the persona lifecycle is an organized collection of processes and tools that will complement other familiar methods. You will use personas to enhance these methods, particularly whenever there is a need for user definition and reference.

In the remainder of this chapter, we will discuss all of the phases of the persona lifecycle, focusing primarily on Conception & Gestation and Adulthood because they are the least understood and, in many ways, the most critical phases in a successful persona effort. Conception & Gestation and Adulthood contain information on how to create personas from data (and/or assumptions, as necessary) and how to use completed personas in the planning, design, evaluation, and release of your products.

Phase 1: Persona Family Planning

Successful persona efforts are ones that are designed to solve specific problems for specific organizations and products. To be successful, you have to know what kinds of problems you want to solve and decide whether or not personas are the best way to solve them. Persona *family planning* is the strategy and planning phase that precedes the creation of the personas.

Three major activities occur during the family planning phase:

1. Researching your own organization (which we call "organizational introspection"),
2. Creating a "persona core team,"
3. Data collection.

Organizational Introspection

Before beginning any persona effort, you (perhaps with the help of the persona core team you assemble) should do some careful thinking about the particular problems you want to solve with your persona effort. While personas can help in many ways over the entire design and development cycle, they cannot solve every problem, and they are not guaranteed to be accepted by the people you feel need them most.

Successful personas are those that meet the needs of *their* users and are built to fit seamlessly into their host environments. In the case of personas, the users are your colleagues, and the environment is your workplace with its existing design and development process.

Ironically, it is easy to forget to turn our analytic eyes on our own users, the people on our teams and in our organizations who use the "products" we produce (e.g., research reports, storyboards, scenarios, prototypes, and other artifacts). We forget to carefully consider who our teammates are, their roles and responsibilities and goals, what is working for them currently and what is not. We launch our bright, fresh, exciting, user-focused ideas into teams who are interested and curious, but who ultimately just need to get their jobs done. As far as most product teams are concerned, they already know the fastest and most effective way to do their jobs; when push comes to shove and deadlines loom closer, your colleagues will inevitably revert to tried-and-true work habits.

Organizational introspection is, in simple terms, working to answer the following questions:

- What resources do we have for personas and other UCD activities?
- What product problems do we want to solve with personas?
- What process problems do we want to solve with personas?
- How can we ensure that the personas will be accepted and used by our colleagues?

Answering these questions now will allow you to decide whether personas will be appropriate and helpful. The answers will help you to create reasonable goals for the persona effort and predict the challenges you are likely to encounter as you create, introduce, facilitate, and maintain your personas. Armed with that information, you will create a plan for your persona effort that will target the application of your personas to appropriate aspects of your development process. Additionally, this information will help you determine what measures of success (and ROI) will be needed at the end of the project.

Creating a Persona Core Team

Personas are a simple idea, but they are inevitably more work than anyone expects. Successful persona efforts involve a core team of between 2 and 10 people who are willing to find time in their calendars to dedicate to both persona creation and to helping evangelize the value and uses of the personas to the rest of the organization.

People who are invited to join core design teams do not necessarily have to understand personas to be helpful; people who

are sensitive to the need for user focus in a company will make excellent core team members.

Your goal is *not* to create a team that will duplicate research or communication efforts; rather, it is to consolidate some aspects of these efforts such that they all contribute to the creation of personas. Plan to include people who are already involved in user research, market research, business analysis, task analysis, or any other user- or customer-focused research or profiling activity. If you have colleagues in any of these specialties, you should put them on the "short list" for inclusion on the core team:

- Information architects, interaction designers, and HCI specialists.
- Usability specialists, user researchers, and ethnographers.
- Technical writers and documentation specialists, and training specialists.
- Market researchers, business analysts, and product managers.

These colleagues are likely to understand the value of personas, both for the organization and for their own projects. They bring with them a deep commitment to user-centered design, experience studying, analyzing, and designing solutions for target users, and an interest in new methods to bring user-focus into the entire organization.

Identifying Data Sources and Collecting Data

From one-on-one interviews to widely published reports, thousands of data resources are available to persona practitioners. Generally speaking, personas are best created from a combination of qualitative and quantitative research. The amount and types of data you will collect will depend on how much time and money you have to spend and your own evaluation of how much data will be necessary to create "good" personas for your project. If warranted and possible, you will do some original user research of your own, though this endeavor tends to be the most time consuming of all.

Most companies have a great deal of data "hanging around." Market research reports, customer segmentation studies, customer service logs, web usage logs and statistics—all of these sources can be reused in your persona effort. In fact, we recommend that you postpone future data collection efforts until after you analyze the data you already have during the Conception & Gestation phase of the persona lifecycle. This process will help you review your existing data sources from a new perspective, and you will be able to target future data collection projects to "fill in the holes" of your current understanding of your users. Look for both internal sources (e.g., market research reports and interviews with product support specialists and other subject-matter experts) and external data sources (e.g., public websites that provide statistical and demographic information).

As you identify and collect data sources,, we also recommend that you identify and evaluate current assumptions about users that exist within your organization. The only assumptions that can harm your product are the ones you are not aware of. We strongly believe that personas are much more credible and helpful if they incorporate and refer to real world data. If data is simply unavailable, or you have no time to collect and analyze the available data, creating *assumption* or ad hoc personas is extremely worthwhile. One of the major organizational benefits of personas is their ability to focus everyone on a shared understanding of who the user is and what he or she really needs out of the product you are designing and building. The key word here is "shared:" everyone in your organization will inevitably build an internal understanding of the users of your product no matter what you do. If, through your efforts, they all picture the same users—even if this picture is built on assumptions—your product will benefit.

Family planning ends at the point at which you have established that (a) personas are right for your organization and current project, (b) you have buy-in from key individuals, (c) you have a persona core team in place, (d) you have a solid plan for the rest of the persona effort that suits your product team's needs, and (e) your initial research and data gathering are complete.

Using Assumption Personas to Help a Multidisciplinary Team See the Need for Personas

Graham Jenkin, User-centered Design and Research Executive, Bank of America ecommerce/ATM

In most companies, a range of team members must come together to define and build customer experiences: product managers, marketers, engineers, researchers, and user experience professionals. While the need for personas may be obvious to user-experience professionals, other team members may require some creative persuasion.

At Bank of America, the eCommerce User-Centered Design & Research (UCDR) team used "assumption" personas to obtain cross-team support for a dedicated persona development effort for its Online Banking product.

Online Banking team members from UCDR, Marketing, and Product Management were interviewed and asked to imagine one consistent customer profile and to make *assumptions* about that customer's relationship to the bank, usage patterns, motivations, goals, needs, frustrations, and attitudes toward technology.

The variation in the team members' assumptions was striking. Some assumed that Casey—the assumption persona, a young, urban professional—had strong loyalty to the Bank of America brand, while others assumed that she had no loyalty at all. Some assumed that Casey used Online Banking for day-to-day financial triage, while others assumed she used it for "big picture" planning. Some assumed that Casey was "geeky and wired," while others assumed she preferred to wait for family and friends to introduce her to new technology.

Of course, no one was right. Team member assumptions were exactly that—assumptions based on intuition and lacking in data.

When the interview results were shared, there was no doubt on the next action. Team members could not agree on who precisely Casey was, but they were unified in agreeing to develop and use a single set of personas based on real customer data.

Phase 2: Persona Conception & Gestation

Persona Conception & Gestation is the process of translating raw data into information and information into personas. A lot of the work during the Conception & Gestation phase centers on collaboratively filtering data and organizing information—both information that arises out of the data collected in family planning and information that arises from other sources, such as inherent knowledge of (a) how people behave, (b) your business or product strategy, (c) the competitive marketplace, and (d) technological affordances related to your product domain.

In this section, we will summarize the process we recommend for getting from raw data to completed personas. This process is designed to work best as a series of collaborative meetings with your core team. The first three steps cover persona conception; steps four through six describe persona gestation.

Step 1: Identify and Assimilate Assumptions

Start with assumptions? After all this emphasis on data? You bet. Your completed personas will incorporate lots of data—but you will be introducing them into an organization that is overflowing with assumptions. If you can, identify categories of users that are important to your business and product domain.

How does your company refer to users today? Every company tends to already have a set of terms they use to describe the relationships and/or differences between their users. These terms are usually collections of defined *user roles, user goals,* and/or *user segments.*

Examples of users defined in terms of segments (e.g., shared demographics or psychograpics):

- A large software manufacturer may think about their users as *small businesses, medium businesses, large businesses,* or *home office users.*
- A travel agency might be used to thinking about their users as *recreational travelers, families,* or *business travelers.*

Examples of users defined in terms of roles (e.g., relationships they have with a system):

- A company that makes online presentation software might think about their users as *audience members, presenters, presentation designers,* and *technical support professionals.*

Examples of users defined in terms of goals (e.g., what they are trying to achieve, in their own terms):

- An online bank might think that the most interesting differences between their customers are that they have different goals: "I have to feel my money is safe," "I have to be able to access my account from anywhere," or "I don't care about anything except low fees."

The goal is simply to identify the ways people in your organization *already* talk about categories of users. Identifying these categories now (even if they are based solely on assumptions) will help you structure your data processing and build a bridge between the ways people think of users today and the data-driven personas you will create. When you are ready to communicate and use your personas, you will find it much easier to do so if you can describe them in language that is already familiar—even in the case where your data suggests that the initial categories should be replaced by different ones.

Get the rest of the assumptions out on the table. Sit down with your core team and lots and lots of yellow sticky notes. Spend 30 minutes or so getting as many assumptions down on sticky notes as you can. Write one assumption on each sticky note. These assumptions should describe what you think your company's users are like as individuals. For the core team working at an online presentation software company, users might include:

- A 46-year-old salesman who lives on the East Coast and has a hard time staying up-to-date with the latest product innovations taking place at the San Francisco home office.
- A CEO who has to give a keynote presentation at a large conference, but has to do so remotely.
- An investor relations specialist who has to present financial data to analysts and stakeholders at least once a quarter.
- A marketing manager who has to make sure that the key marketing messages for her company are distributed to the right people at the right times.

After everyone has created as many sticky notes as they can, it is time to find patterns in the assumptions. To do this, you will conduct an affinity exercise. Write the major categories of users (the ones identified in Step 1) far apart from each other on a large sheet of paper. Ask everyone to place their sticky notes on the paper so that sticky notes with similar or related assumptions are near the appropriate category and near each other, and dissimilar assumptions are far apart. This exercise should be a noisy, collaborative experience as members of the core team discuss placement and groupings of their sticky notes.

If some sticky notes do not relate to any of the listed categories, create new areas on your large sheet of paper for them. As the exercise progresses, groupings of related sticky notes start to form "clusters." When the team feels that they are finished clustering, ask them to label the clusters. For example, one cluster could be labeled "Marketing People," and another, elsewhere on the big sheet of paper, could be labeled "Investor Relations People."

Discuss what you have found as a group. Are there any surprises? This is a great time to think about the kinds of information you would like to find in your data, or collect directly from users. Your data will validate and enrich the categories, or it will provide solid information to show that the existing categories are inappropriate. It will also allow you to define important subcategories of users that should also be expressed in personas.

Step 2: Process Your Data

It is time to process the raw data and extract information relevant to your user- and product-domains, and then identify themes and relationships. While you can process your data in

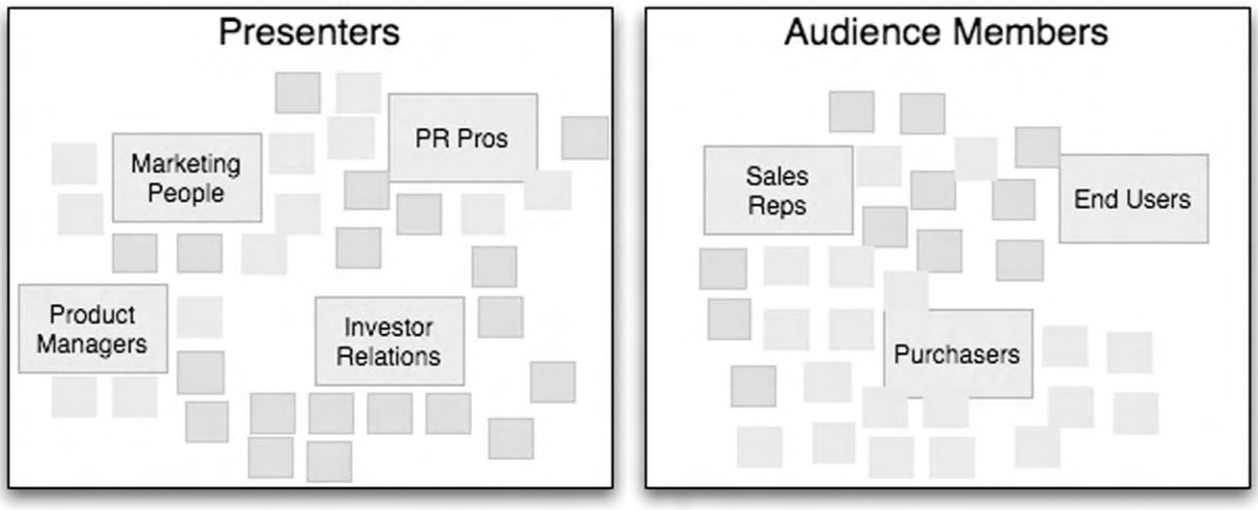


FIGURE 51.2. An example of clustered assumptions (light gray stickies) and factoids (dark gray stickies) with cluster labels (large stickies) and the initial categories of users. Notice that there are clusters that contain only assumptions or factoids in addition to clusters that contain both.

many different ways to create personas, we strongly recommend that you conduct an affinity exercise like the one you have just done with assumptions. Before you get started, assign a number to each of your data sources.

Identify key data points (factoids) in the data sources. The first step in processing the data is to consume and filter the information that is in each of the data sources. You will want do this because not every data point in a given study/report is relevant to the definition of your target audience or the design of your product. Whether it is done before or during the meeting, ask your core team members to highlight findings that they think are key to understanding your target audience, or that are highly insightful toward defining aspects of your product. In other words, you want them to look for findings that are relevant to your market, industry or domain. Highlight any facts that seem important to your product's audience or to the product itself.

Each important factoid should be copied or cut out of the original document. Remember to note the source and page number on each factoid so you can trace them back to the original sources later. We recommend that you use blue sticky notes for this exercise. Whatever color you use, don't use the same color that you used for the assumption stickies; you will need to distinguish between the assumptions and the factoids.

Asssimilate the factoids. Now the interaction (and fun) begins. Everyone should add their factoids to the clusters that were created in the assumption exercise. Don't be afraid to move any of the stickies around as you do this—you can move assumptions, factoids, or even entire clusters as you assimilate.

As you continue assimilating, keep your eyes out for large clusters, or "puddles," of 8–10 or more sticky notes. Large clusters can usually be broken down further, and you should do this if you can. Usually, we recommend that everyone find spots for their stickies and then pair up to review the clusters together. During this "clean up" phase, you will be able to break up puddles, add descriptive labels to clusters, and generally ensure that the clustering makes sense. Stop when the activity dies down and few stickies are still being moved.

Note that you will have clusters that contain both assumptions and factoids, but you will probably also find clusters of assumptions (with no factoids) and clusters of factoids (with no assumptions). This is illustrated in Fig. 51.2. All three types of clusters are helpful. Clusters of assumptions without factoids will help you identify topic areas in which you need to collect more data. Clusters of factoids without assumptions tell you that there are aspects of your customers that you have not thought a lot about yet.

Another Perspective

The Cooper Method: Collect Data Directly From Users and Identify Patterns to Create Personas

Kim Goodwin, Vice President and General Manager, Cooper

Some people have great success with personas, while others do not. Why is that? First, it is important to understand that even the best personas will not solve all of your problems—scenarios, design judgment, and visualization skill are equally important. Assuming you have all of those things, the key to success with personas is to do the right kind of research and to make sure your personas truly reflect your findings.

1. Start with the right kind of research

Personas that are made up without data are not really personas, and although they can still be useful thought exercises, they are far less effective than real personas that are

based primarily on ethnographic user data. Ethnographic techniques are valuable because they assume that an interview subject's attitudes and behaviors are so habitual as to be unconscious. Rather than asking users what they want, it is more effective to focus on what users do, what frustrates them, and what gives them satisfaction. By combining interviewing with direct observation—preferably in the actual usage context—you can get a lot of data very quickly. Observation also helps minimize dependence on users' self-reported behavior, which is often inaccurate.

At Cooper, we send the designers out to do the research, so they see the problems firsthand and develop empathy with the users. We may occasionally have three interviewers, but two of them are consistent across all of the interviews; this makes synthesis much easier later on. We spend 45 minutes to an hour with individual people. It would be easy to write a whole chapter on research techniques alone, but there are a few fundamental points. First, ask a very broad question (such as "Could you think of a typical work day recently, and walk me through it?") This raises a number of issues the interviewers can pursue without needing to ask leading questions. Another important technique is to stay case-focused; in other words, ask for specific instances rather than generalizations. This will get you more detailed and more accurate information. Also, be sure to look at and ask questions about artifacts or aspects of the environment. Finally, focus on actual behavior and frustrations, rather than asking the users to design the product for you.

Contrary to some expectations, this kind of research can be done in anywhere from a few days to a few weeks. You will know you can stop interviewing when you can predict how each user will respond; this means patterns are beginning to emerge. If you have the time and budget, you can verify your findings with quantitative surveys or other techniques, but these cannot replace direct observation.

2. Identify behavioral patterns from the data

Once you finish interviewing, list all of the behavioral variables for each user role (e.g., ways in which interviewee behavior differed). In an online shopping domain, for example, you might have variables such as frequency of shopping, degree of enjoyment, and price versus service orientation. You may find that some demographic variables—such as age or technical skill—seem to affect behavior. Be wary of focusing on demographics during persona creation, since behavioral variables will have far more impact on the design. Note that if you are doing an enterprise application, each role will have its own set of behavioral and demographic variables. Although the number of variables will differ from project to project, it is typical to find 20 or so variables per role. Map each interviewee against the appropriate set of variables, and then look for people who clump together across a large number of the variables. When you have found a set of people clustering across six or eight variables, there is a good chance that you have found a major behavior pattern that will form the basis of a persona.

3. Turn your patterns into persona descriptions

For each pattern, add details based on your data. Describe the current potential usage environment, typical workday (or other relevant time period), current solutions and frustrations, relevant relationships with others, and goals. Avoid the temptation to add a lot of irrelevant personal detail; if you are designing an email tool, it does not matter that your persona wants to be an actress. One or two bits of personality can bring an otherwise dull persona to life, but too much biography will be distracting and will make the persona less credible as an analytical tool. If every aspect of the description cannot be tied back to real data, it is not a persona—it is a creative writing project that should not be used to make critical design and business decisions. Describe each persona in a one or two page narrative that helps stakeholders understand what makes them tick.

4. Use the personas to drive scenarios, requirements, design . . . and communication

Use your personas to develop scenarios; put the personas in realistic future situations and envision how they would like a magic black box of a product to work. These scenarios will lead you to a set of needs you can discuss with stakeholders. You can describe requirements from the personas' points of view, which leads to less resistance than requirements that come from you. Once there is agreement on the requirements, additional scenarios (along with good design skills!) will help create the conceptual framework for the design. When you illustrate the design direction in a scenario, stakeholders are more likely to see the value of the solution. Personas will help you all the way to pixels and specifications, and even through implementation.

Step 3: Identify Subcategories of Users and Create Skeletons

Look at the data clustered under each of your user categories. As a team, evaluate and discuss the possibility that each category should be divided into two or more subcategories. Consider roles, goals, and segments in this assessment. As you identify subcategories, you can write them on a whiteboard; you may also find it helpful to transfer the subcategory names onto sticky notes and place them appropriately in your assimilated data. In doing this exercise, you are simply exploring the possible groups of users that have emerged from your data. Try to identify "differences that make a difference" within each category, based both on the clustered assumptions and the clustered factoids.

Create skeletons. Once you have identified and agreed upon the categories and subcategories of users, you are ready to create skeletons. Skeletons are very brief, usually bulleted lists of distinguishing data ranges for each subcategory of user. Skeletons help your core team transition from thinking about

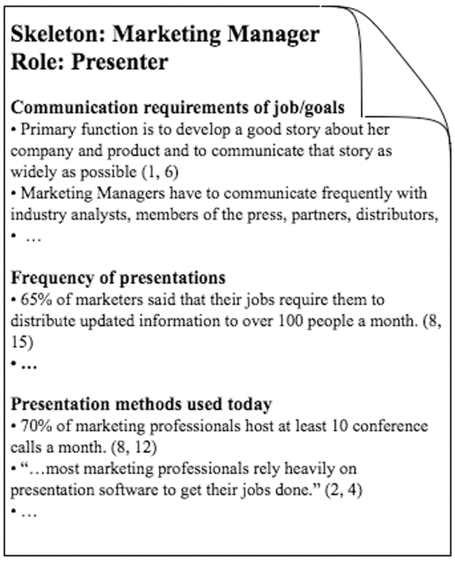


FIGURE 51.3. An example of a skeleton created out of assimilated data. Note that the skeleton includes factoids grouped according to topic; it does *not* include any narrative details or "personalized" information. The numbers refer to the data source number and page number.

categories of users to focusing on specific details; they also allow your team to present the key findings of the assimilation exercise to stakeholders.

Create one skeleton for each of the subcategories you identified. On each skeleton, list the cluster labels that relate to that subcategory; these cluster labels will become headings in your skeleton (see Fig. 51.3). Because you will be comparing skeletons to and prioritizing skeletons against each other, it is important that each one contain at least somewhat comparable information. Consider including common characteristics or headings across all of your skeletons. (If you do this, you may find that you are missing information for some skeletons. In those cases, either leave that information blank, perhaps marking it as "need data," or make an informed estimation about what it might be. If you do the latter, be sure to indicate that it is an assumption to be followed up on.)

Feel free to create as many skeletons as you and your team feel are necessary to "cover" the discoveries you made during the clustering processes. You will have the opportunity to combine and prioritize them in the next steps.

Sometimes Categories of Users (and Personas) Should be Based on Verbs, Not Nouns

Karen McGrane, Vice President, User Experience, Avenue A | Razorfish

Though the method of developing personas is relatively new, it seems a given that a persona maps to a person. The approach taken by user-centered designers focuses on understanding the needs, goals, and mindsets of a given individual. A persona usually describes a single, stable individual whose interaction with a company, product, or website stays consistent throughout the experience.

What happens when the goals and mindsets of an individual user can change rapidly over the course of their relationship with a company—or even over the course of a single session? When we observed people visiting the *New York Times* website, we realized that individual users did not fall neatly into a single mode of use. People tended to switch their goals over time. People used the site differently in the morning, when they wanted to read headlines and catch up on the day's news, as compared to the afternoon, when they might be looking to take a break over lunch or between meetings. We often observed people change modes in the course of a single session, moving fluidly from a news-reading mode, scanning headlines, into a planning mode, where they would research restaurants and read movie reviews.

When it came time to document our personas, we realized that the standard way of documenting them would be limiting. We did not want to imply that "Ketan, a 54-year-old technology consultant" was our only "news junkie," or that "Lisa, a 32-year-old account executive" would always operate in a "planning mode." Making a one-to-one connection between the person and the action seemed inaccurate, since we knew each person would most certainly take multiple actions during their use of the site.

What's more, putting the emphasis on the noun (the persona) rather than the verb (the activity) did not really help the *New York Time*s achieve their business goals. While they absolutely wanted to create a usable experience for a visitor operating within a given mode, they also wanted to know what would prompt someone to change modes, since shifting modes extends the experience to more page views and more frequent visits. As experience designers, we concluded that we didn't need to focus on the person(a.) We needed to focus on the activity—the verb, the action, the mode of use (see Fig. 51.4).

Clients are eager understand their customers better, and many focus on personas as the tool that will help them do that. In the future, I will encourage my clients to think about other means of understanding customer behavior. I think one of the most important ways they can do this—outside of personas—is to learn more about customer goals and modes of use.

Step 4: Prioritize the Skeletons

Once you have a set of skeletons, it's time to review them in preparation to get feedback from all stakeholders. When you meet with stakeholders, you will evaluate the importance of each skeleton to your business and product strategy, and prioritize the skeletons accordingly. The stakeholders will help to identify a subset of skeletons to develop into personas.

Before you meet with the stakeholders, it is a good idea to do some initial prioritization of the skeletons. The core team should carefully review the skeletons you have created and

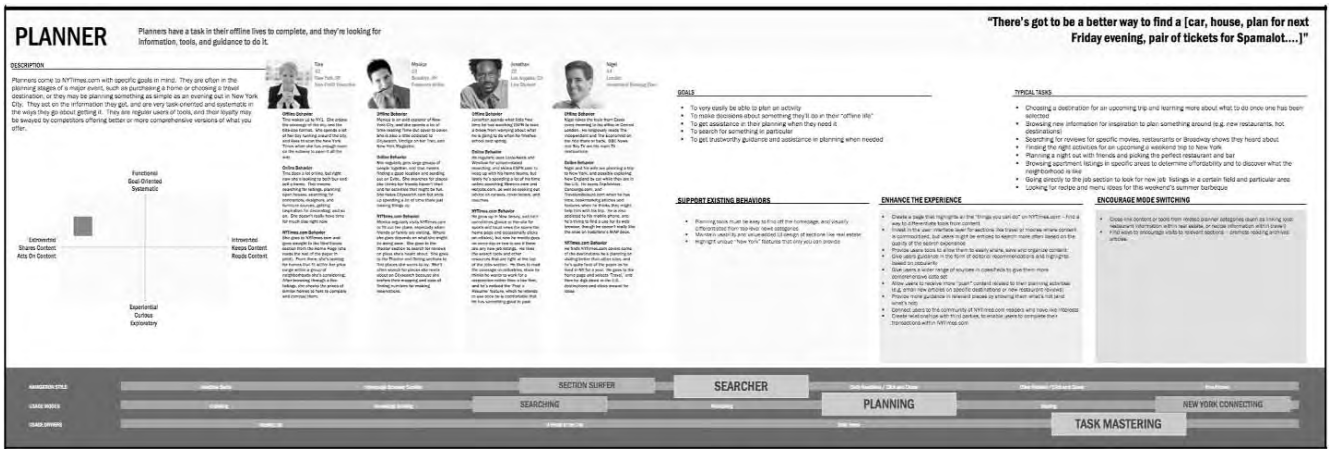


FIGURE 51.4. One of the "verb-based" persona posters for the New York Times project. Note that the poster is focused on the word P*lanner* and not on a specific persona name. In fact, the "Planner" poster includes details about four individual personas.

make sure you agree that each one truly does reflect a "difference that makes a difference" between subcategories of users as identified earlier. In many cases, it will be possible to reduce the number of skeletons by combining several of them. Once you have a set that you feel should not be further reduced, prioritize them in a way that makes sense to the core team. This will give the stakeholders as "strawman" to work with—which is easier than asking them to prioritize a series of skeletons from scratch. As you do this, think about the following questions:

- Is this category of subcategory important to our product (relevant, unique, illuminating)
- How important is it to our business?
- Are there any groups missing?
- Are some of the categories almost right, but a few of the characteristics are "off" and need tweaking?

Note that you should prioritize skeletons *within* each major category of users. For example, prioritize all the "business traveler" skeletons relative to one another, and the "families" skeletons separately. Eventually, you will want to know the priorities of your personas *across* these categories, but first, you will need to narrow in on the correct skeletons *within* each category.

Now schedule a meeting with stakeholders empowered to make decisions about the strategic focus of the company. If stakeholders are not aware of the data and general process that led to these skeletons, present that information before introducing the skeletons to them. Introduce the skeletons in one category at a time, and ask them to assign priorities based on:

- **Frequency of use:** How often would each skeleton use your product? Daily users would likely be more important regarding design decisions than those that only use your product once a month.
- **Size of market:** Roughly how many people does each skeleton represent? Larger markets are usually more important than smaller ones. Do you plan to aim your new product at a new market? In that case, you might consider the importance of a small market with growth potential.
- **Historic or potential revenue:** How much purchasing power does each skeleton encompass? If this is a new product, you may have to estimate this amount (e.g., through trade journals, market trends, market research, and understanding spending behaviors in different by related markets). In many cases, users might not directly make the purchase; someone else buys such products for them. Still, they may influence those purchase decisions.
- **Strategic importance:** Decide who is your most strategically importance audience. Is it (a) those who make the most support calls, (b) those that rely on your product for critical activities, (c) those that use your competitor's product, or (d) those that don't use yours or anyone's product yet? Are you trying to expand or grow your market? If this is your primary goal, do your skeletons include nonusers, technology pioneers, or trend setters? Which target audiences will help your team innovate or stretch?

Prioritization can be difficult, and your first meetings will likely end in requests for more data. For example, stakeholders might ask you to find data on the market size for each of the skeletons before they feel comfortable prioritizing. If prioritization feels impossible, it might mean that the business goals are not clear. This can lead to some delicate conversations. In general, remind the stakeholders that you want to prioritize the skeletons so that they accurately reflect the company's vision for itself and the product. The stakeholders *must* be involved in the prioritization of the skeletons, because they are the ones who are setting the course for the business. Be willing to go back and try to find more data if necessary.

Step 5: Develop Selected Skeletons Into Personas

You now have a reduced set of basic skeletons. Your task at this point is to enrich these skeletons to become personas

by adding additional data, as well as concrete and individualized details to give them personality and context. You'll also include some storytelling elements and photos to make them come to life.

As you build on your skeletons, all of the details of your personas will be encapsulated in a "foundation" document (see Fig. 51.5). Foundation documents contain the complete definition of a given persona, but they do not have to be long or difficult to create. Depending on your goals and the needs of your team, your foundation document could range from a single page to a long document. Creating a foundation document for each persona will provide you and your team with a single resource you can harvest as necessary while creating your persona communication materials. At the very least, complete personas must include core information essential to defining the persona—the goals, roles, behaviors, segment, environment and typical activities that make the persona solid, rich and unique—and more importantly, relevant to the design of your product.

If you are extremely time- and resource-constrained, you can start with brief one-page or resume-style foundation documents. Then, as you find the time, you can always come back and add to the information in these short foundation documents.

When you are deciding which characteristics to include in your foundation documents, think about the types of information that will be most helpful to your core team and to the development team. We recommend you include at least rudimentary information in each of the following categories:

- Photo(s)
- Name
- Age
- Goals/motives/concerns
- Personal details/family life
- Income/Spending habits
- Work/job details
- Use Environment/artifacts
- Activities/use scenario
- Knowledge/skills/abilities
- Likes/dislikes
- Quotes
- Market Size/Influence

To create the content of these sections, you will turn the characteristics in your skeleton personas into very specific, more concrete values. These specific details turn your collection of facts into representations of people. For example,

- High-Tech, publicly traded companies *becomes* "PrinterCo."
- Works in a communications role *becomes* Marketing Manager.
- 70% Female *becomes* Megan, Dianne, Irene, and so forth.
- Lives in a major metropolitan city *becomes* Chicago, LA, or Houston.

As you replace factoids with specific details to enrich your persona, copy the factoid or set of factoids into a comment or a footnote in your foundation document. A lofty but worthy goal is to have every statement in your Foundation Document supported by user data. You will not likely achieve this, but the attempt helps you to think critically about your details, and highlights places where you might want to do further research.

Moving toward precision means moving away from accuracy. In many cases, the accuracy of your data lies in its ranges (not just central tendencies, but descriptors of variance, percentages, and skew), and by selecting precise descriptors, you are going to lose some of that accuracy. For example, if a category includes males and females, you cannot create a single individual who "represents" the entire category. Rather than trying to represent every nuance of the entire category, try to pick values that are reasonable, believable, and meaningful.

Think of your data, and your categories and subcategories of users, as describing *neighborhoods* of related users of your product. As you create your personas, you are describing a single, specific "resident" of each neighborhood; as in real life, each resident inhabits his or her neighborhood, but no one resident can represent all the qualities of all the people in the neighborhood.

Incorporate narrative and storytelling elements. Enriching your terse skeletons into personas that are realistic and engaging requires some storytelling. To do this well, consider that you are trying to "tell the story" of the data in your foundation documents with narrative. How do your personas sound and act? What can they do or not do? Turn your factoids and specific details into a running story—a sequence of actions and events with interaction and a plot. Demonstrate their interactions with people, objects, and systems. In the best case, these stories are modeled on real, representative cases revealed through qualitative research.

Narratives in persona documents are usually written in third person, active voice. Here is an example of a beginning descriptive overview for a marketing manager named Megan written as a narrative:

> Megan is the product marketing manager for PrinterCo, a leading printer manufacturer. Part of Megan's job is to create and foster the company's image as a cutting-edge technology company. Her primary function is to develop a good story about her company and product, and to communicate that story as widely as possible. She spends her time thinking about how to educate people about PrinterCo's products and to build relationships between customers and PrinterCo.

Be careful when evoking stereotypes or any information that could elicit a strong personal response. When in doubt, choose to include details that help others see your persona as a real person, with particular and understandable goals, needs, and interests. Allow realism to win out over political correctness; avoid casting strongly against expectations if it will undermine credibility. Break the mold if it helps get people on board with your effort. Alan Cooper (1999) addressed this issue by stating, "all things being equal, I will use people of different races, genders, nationalities, and colors" (p. 128).

Illustrate your personas. Each persona needs a face, or a photo or set of photos, to make them real. We believe photos or illustrations are critical; they help your team believe in the

personas and understand that each persona describes a single person. The choice of what specific photos to use is a hard one. These illustrations of your personas are extremely influential and can significantly affect how your personas are perceived.

A photo is more than just a face; the model's clothing, expression, activity, and general appearance along with the setting and background will all communicate or dictate some of the characteristics of your persona. You can either take advantage of this fact or continually fight it. In general:

- Avoid "slick" stock photos. Photos of models *look* like photos of models—not of real, everyday people. A great alternative is to take photos of friends-of-friends. Do not use photos of a person that anyone on the product team knows. If you do take your own photos, take many! You can use different photos at different times to keep people interested in the personas.
- If you cannot take your own photos, look for photos online. Flickr.com is a great example of an online photo-sharing site, and you can look in the Flickr "Creative Commons" area for work by photographers who are willing to let others use their photos royalty-free. Do a search for "portraits" to find hundreds of potential personas!

If possible, include multiple photos in your foundation documents to illustrate your persona.

Step 6: Validate Your Personas

Once you have added details, it is important to double-check to make sure your final personas still reflect your data. Your goal is to ensure that you did not stray too far away from your data when you made their characteristics specific and concrete and added elements of story telling. While it is true that personas cannot and do not need to be completely accurate, you do want to ensure that they reflect the essential information about your target users that you found in your data. If you built assumption personas, you should ensure that the personas you created really do capture the assumptions in your organization.



FIGURE 51.5. A portion of an example persona foundation document. Note that the callouts on the side of each page are "factoid" references.

To validate your personas, you can do one or more of the following:

- Review your personas against the original data sources.
- Have experts, those that are closest to your users, review your personas.
- Have representative users of each persona review "their" persona.
- Conduct "reality check" site visits.
- Conduct large sample surveys or interviews and apply statistical analysis or modeling.

These five approaches are not mutually exclusive, nor are they the only means to validate your personas. Treat the validation process as an opportunity to gather even *more* data about your users, incorporating significant findings back into the persona definitions.

Completed personas do not mark the end of user research. When you finish the creation of your personas, you may be tempted to think that you do not need to further understand (do research on) or involve real users in the development of your product. From our perspective, this could not be further from the truth. We believe personas are a great starting point for understanding and incorporating user information in the development cycle.

Phase 3: Persona Birth & Maturation

Personas are typically created and communicated as static documents or posters that provide a snapshot of interesting and relevant information about users. In fact, the foundation document we describe above is a case in point (see Fig. 51.5). These artifacts have proven helpful, largely because they help make information about users highly accessible, engaging, and memorable to people making decisions. Such representations of users, however, are not "alive." They are depicted as motionless portraits, usually contained within a single, finite document and presented as such. There is no room for growth or development. That is, unlike a character in a book or film, such descriptions do not evolve. Moreover, the team using them is supposed to "get to know them" almost instantly. When we get to know a friend, neighbor, colleague, or even a character in a favorite book or TV show, we *build up* an understanding of them, a relationship with them. Once we know people, we are able to understand why they do what they do, what they want, and what they need. Engendering this level of understanding is the next frontier for user representation.

We believe you must enable the personas to "come to life," allowing them to be alive and to develop in the minds of the people using them. Toward this end, we propose that persona practitioners must

- Embrace the challenge of communicating information about users through narrative and storytelling.

- Incorporate a variety of formats and media to communicate the essential persona characteristics.
- Maintain a lifecycle perspective when educating colleagues about the personas.
- Allow the people using the personas to extrapolate from and extend them.

In other words, personas should be more than a static collection of facts. Personas are compelling stories that unfold over time. To be very clear, we are *not* suggesting that personas change drastically over time, take on new characteristics, or develop new skills; they are not to be moving targets. We believe that successful personas and persona efforts are built progressively; just as we get to know people in our lives, we must get to know personas (and the data they contain) by developing a relationship with them. No single document, read in a few minutes or posted on a wall, can promote the kind of rich, evolving relationship with information about users that is the cornerstone of good product development. No single document can contain the wave of scenarios and stories that your personas will inspire. Personas must be aided, if they are to *live in the minds of your colleagues*. As long as the personas are well built, data-driven, and thoughtfully communicated, the product team can use the personas that come to exist to generate new insights and seek out the right details when they need them.

Like parents sending young children off to school, you and your core team will send your personas into your organization to interact with other people. The personas are fully formed but may continue to evolve as your team becomes familiar with them. Problems at this phase can lead to a lack of acceptance or visibility or personas that "die on the vine" and disappear from the project. More subtly, your personas may come to be misconstrued and misinterpreted. Successful persona Birth & Maturation requires a strong, clear focus on communication to ensure that your personas are not just known and understood, but adopted, remembered, and used by the product team. The Birth & Maturation process includes

- Creating a persona campaign plan to organize your work in Birth & Maturation and Adulthood.
- Introducing the personas (and the persona method) to the product team.
- Ensuring that the personas are understood, revered, and likely to be used (e.g., creating artifacts to progressively disclose persona details).
- Managing the minor changes to the persona descriptions that become necessary after the personas are introduced.

At this phase, you must be prepared to answer the difficult questions that will inevitably come up as you introduce the personas. You will have to be prepared to discuss (a) the process you used to create the personas, (b) their utility, (c) the ways you would like the product team to use the personas, and (d) the ways you intend to measure the value of the persona effort. The work you did in the family planning phase will come in quite handy as you prepare your answers to these inevitable questions.

Using Personas to Build Teams

Aviva Rosenstein, Manager, Design Research, Yahoo! Media Group

The Yahoo Photos team supports an application for sharing, printing and organizing personal photos. This team has been in the vanguard in adopting Agile development approaches at Yahoo as an alternative to the traditional waterfall development process. To make this transition to an Agile approach successful, we needed to find ways to bring engineers and designers together into a single, cohesive team.

When introducing the persona set to the project team, I used our design personas to create connections between the engineering and design staff while, at the same time, familiarizing the entire team with our target customers, by creating a simple game requiring knowledge of each of the personas' characteristics. The design team scheduled a happy hour to introduce the personas, created handouts and posters with information about each persona, and invited the entire product team, including all the designers, developers and product managers assigned to the property, to come play the game (see Fig. 51.6).

The exercise was a success; everyone had to study the persona artifacts to determine what questions they could ask, which gave them a head start in internalizing our personas. In addition, the game provided an enjoyable context for designers and engineers to cross social barriers and begin interacting with each other.

The persona artifacts served as "boundary objects" (Star & Griesemer, 1989), a way of bridging communication gaps between disparate functional and organizational worlds. The exercise itself broke down the barriers that existed between the engineering and design groups and allowed the design team to get the rest of the developers to empathize with the needs of our customers, but the actual impact of the persona artifacts went even further. After this exercise, team members commonly and naturally began to use our personas to refer to actual customers and their needs, in storyboarding, use cases, and requirements documentation. Project management used the personas to communicate value propositions with executive staff. Marketing used the persona artifacts to communicate target audience characteristics to our outside PR agency. Design management used the handouts to communicate requirements to an outside design firm retained to develop a specific part of the service. It gave all of the teams a shared language for talking about our business. In addition, introducing the persona set in an informal and engaging way encouraged adoption among the various disciplines contributing to the overall success of the project.

Persona Artifacts

You can use a variety of methods to communicate person as to the members of your product team, including websites, posters, illustrations, Word documents, Visio diagrams, live actors, and



FIGURE 51.6. The Yahoo! Photo persona artifacts helped bridged a communication gap between functional and organizational worlds.

videos. Remember that the artifacts and materials you create to communicate information about personas are very important—they are the user interface for your personas and the data behind them. Well-thought-out and well-designed persona materials can add credibility to your entire persona effort and help enormously with your persona communication campaign.

We define three major categories of persona artifacts:

1. **Buzz Generators**. These artifacts should be designed to build up anticipation about the introduction of the personas. They are usually posters with relatively little information on them. Buzz generators can give hints about the fact that personas are coming (e.g., a poster that says "do you know who your users are?") or begin to introduce the personas themselves (e.g., a wanted-style poster showing just a persona's photo, name, and perhaps role: "Meet Barry the Business Traveler").
2. **Comparison Facilitators.** Comparison facilitators are helpful after everyone has been introduced the personas. These artifacts should be designed to help people understand key differences between the personas. For example, you could create a table-style poster that lists the personas across the top and highlights different goals, technical abilities, challenges, and so forth across each row.
3. **Enrichers.** As the design and development process continues, you can use enricher artifacts to refresh everyone on specific aspects of each persona. For example, when the team begins working on security features, you could create enricher artifacts that describe the security challenges for each of the personas. You might also send out a monthly e-mail "from" each persona to update the team on their goals and needs. You can also create fun artifacts as enrichers; for example, you can hand out candy bars that have new "persona" labels listing salient facts about each persona. Anything that helps keep the personas fresh and alive in the minds of the team is a good thing!

Note that it does not take a lot of these artifacts to have an effective communication campaign. Be very strategic and frugal in your choices. Approach the creation and distribution of your persona artifacts carefully:

Agree on the specific goal of the artifact. Why are you creating this specific artifact? The goal will probably be related to one of the three categories of artifacts we previously described.

Agree on the audience, timing, and distribution method for the artifact. Your persona artifacts should eventually be everywhere around your office (in hallways, coffee room, doors, meeting rooms, stakeholders/leaders' offices, etc.), but they should appear progressively. For every artifact, consider who is going to see it, when (in the development cycle) they are going to see it, and how the environment will affect their ability to digest the information. For example, you might decide to create different "buzz generator" posters for the developers' hallway versus the marketers' hallway. If you work in place that does not allow posters and such to be displayed around the building, create artifacts that can handed out to individuals, carried around, or placed on desktops.

Agree on the information elements that should (and should not) be included on the artifact. By the time you are ready to create persona artifacts, you will have quite a bit of information about each persona at your disposal. The information you have will all seem highly relevant and deeply interrelated, and it can therefore be difficult to comb out small snippets to include on individual artifacts. Remember that the easiest way to create a useless persona artifact is to overload it with in-

formation. For example, you might decide to create "wanted" posters to create buzz and to convey the name, role, and picture for each of your primary personas. It will be tempting to include a quote and maybe a few bulleted details with additional information. Remember your priorities; if you really do want to build buzz and interest, consider limiting the poster to just a photo, a name, and a role. When in doubt, always opt for less information and leave your audience craving more.

Agree on the relative priorities of the information elements on the artifact. Once you decide which information elements should be included on an artifact, prioritize these elements according to how important it is that the element is read and understood. For example, on the "wanted" posters, the photo and name should probably be very large and eye-catching. In contrast, a comparison poster you distribute a few weeks or months later should include names and roles, but these are probably not as important as comparative information about each persona's goals, abilities, desires, and so forth. See Fig. 51.7 for example posters in each of the three categories of artifacts.

If you have limited resources (e.g., very little money to use on persona artifacts), think carefully about the artifacts you will need now and try to predict what you will need later. Do not use your entire budget on artifacts you will distribute early; remember that you still face the challenge of keeping the personas alive and useful throughout the Adulthood phase.

Phase 4: Persona Adulthood

Personas are "all grown up" in the adulthood phase, and they have a job to do. You have now introduced the personas to the product team and worked to clarify the role and importance of the personas. You have encouraged the product team to embrace the personas and the information they contain, and now it is time to help everyone use the personas to inform the design and development of the product.

Adult personas are ready to be put to work in a variety of ways. Personas can help by answering difficult questions and by focusing activities in a way that takes the guesswork out of making customer-driven decisions. Personas can participate in your product planning, design, and development process by

- Being present at your meetings and representing the voice of your customer throughout the development cycle.
- Providing consistency by providing a common reference point across your organization, even in a highly chaotic, fast-moving, ever-changing environment.
- Providing a way for all of the product teams to touch base using a common language, and to ensure that everyone is staying focused on creating a good experience on the right audiences.

Personas can only be involved and helpful in these ways if they inhabit your workspaces and attend your meetings. Even though they are not real people, personas can become the most powerful voices in the room. Large wall posters, which introduce your personas, should be placed in every meeting room as well

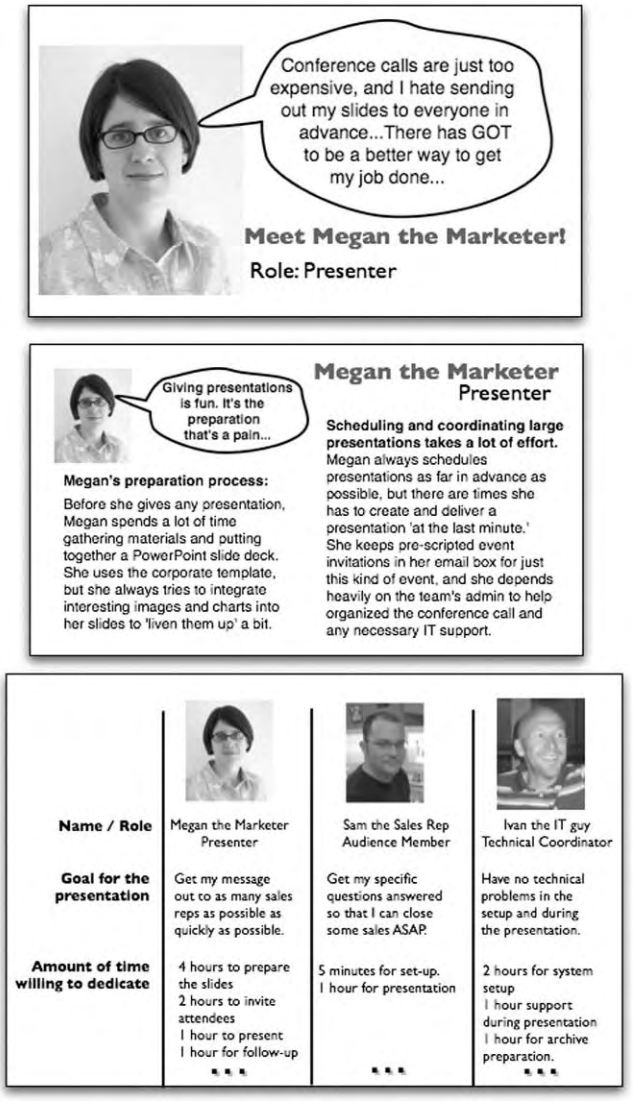


FIGURE 51.7. Examples of a buzz generator, and enricher, and a comparison facilitator poster for the online presentation software personas.

as other common spaces where team members discuss product design, features, and overall user experience. The important thing is that the personas' presence is felt. You want your teammates to refer to the personas in their everyday work. Hundreds of tiny decisions are made each day, and you want these decisions to be made in consideration of your target audiences.

The more the persona names are used, the more likely it is that everyone in your organization will accept the fact that the personas and the user data they represent are here to stay. The more the persona names replace the word *user* in documents and conversation, the more likely it is that the persona data will shape your product. We suggest that you ban the word *user* across your organization. Find the owner of each and every document that will help to define or describe your product and encourage them to use the names of the personas instead of the

word *user*. You have succeeded if you can search the Product Vision Statement, Business Plan, Technical Spec, Marketing and Messaging Plans and find no instance of the word *user*.

Use Personas for Product Planning, Design, Evaluation, and Release

One obvious way to use personas is to "test" design ideas by asking questions such as, "Will Sally want to use this feature?" We have found that this is only one of the many ways personas can be involved in the product design and development process. The effective persona practitioner must understand the many other ways personas can be involved in existing processes, and ensure that the personas work hard in an organization during the entire development process.

Personas can be used to (a) inform quality assurance test cases, (b) recruit usability test participants, (c) make high level and detailed feature priority decisions, and (d) communicate product direction. Personas can also inform marketing, advertising, and sales strategy.

Personas and product planning. Personas can help your team envision what your new product should do, understand the context into which you will launch your product and the kinds of problems it needs to solve if it is going to be successful. Now that you've created your personas, you can ask the personas to "tell you their stories;" the needs, goals, and contexts you so carefully included in your persona descriptions will now allow you to generate helpful stories about the way your product will be used and the actions (and reactions) it should elicit.

Personas can help in the product planning process by helping you both discover important features and evaluate the relative values of each feature. You can use personas to help you understand and capture your user and system requirements through

- Persona narratives and storytelling.
- Persona-focused competitive reviews.
- Persona-focused feature brainstorming.
- Persona-focused evaluation of proposed features.

All of these involve taking on the perspective of your personas to review the competition, your ideas for features, and so forth. For example, you can do a persona-focused competitive review by finding out with which existing products your new product will compete. Your marketing team has probably already done this, and they are a good source for help with this exercise. If you can, buy a copy of each competing product. (If the products are prohibitively expensive, you can do this exercise using the marketing or collateral materials instead of the actual products; this will give you insight into the reaction of your potential customers to the messages your competitors have deemed to be important.) Once you have access to the products, it is relatively easy to look at them from your personas' perspectives.

Gather the core team and ask one member to "walk through" the competitive products from the perspective of one of you personas. As you observe your "persona" walking through the product, you will find aspects of the product that work well and some that do not. If members of your product design team are present, they will come up with ideas for functionality that you must address in your product and ideas for brand new features.

Personas and product design. Once your organization has a vision and overall development plan in place, it is time to design the elements of your product. Your personas helped you understand the big picture, and now they can help you make decisions about specific features and design elements—that is, your personas can show you how these feature appear and behave.

We have found one process to be particularly helpful in translating insights derived from personas into ideas for product designs. It is called Design Mapping. We will describe this method in depth in the next section of this chapter so that you can try it out for yourself.

Personas and product design evaluation. As your team settles in on the features and specific solutions that it needs to embrace to create a successful product, your personas can help in honing the implementation of these features toward the very best design.

You can use your personas to help with evaluation of your features and solutions through

- Cognitive walkthroughs and design reviews with personas.
- User testing and ongoing user research with persona profiles.
- Quality assurance (QA) testing and bug bashes.

The best way to incorporate personas into the above processes is to involve the owner of each process (for example, the QA manager) to meet with the persona core team. As a rule, meet with process owners earlier than you think is necessary. This will give you time to figure out ways to incorporate personas into the various evaluation processes your company uses before it releases any product.

Personas and product release. Now that your product is getting close to being complete, it is time to turn your attention toward details that are not directly related to product development. You have put a great deal of effort into creating and using your personas to design and build your product. Now that the product is almost complete, your personas (and all of the persona-related materials and tools you have created) can be extremely helpful to those responsible for documenting, supporting, and selling your product.

If you have used the personas throughout the design and development process, you will have many documents that talk about how the product should to work from the point of view of the personas. These documents will be invaluable to the documentation, training, and support professionals in your organization. Marketing and sales professionals will also be able to use the personas (and related deliverables) to help craft materials to support their own work.

Many persona practitioners have told us that the personas seem to "move out" of the design and development offices and "move in" to the documentation, support, and sales offices during product release.

A Great Tool for Persona Adulthood: Using Design Maps to get From Personas to Product Designs

Design Mapping is a process that results in a large flowchart created out of sticky notes, and depicts an individual's end-to-end experience of using a tool or accomplishing a goal. Maps are artifacts that help you understand and communicate information about the ways people achieve their goals and the ways they could achieve their goals with new tools. Maps are similar to other participatory design tools, but are useful in ways that we have not found other tools to be. Design Maps tell stories about the *experiences* of personas in the future.

Maps are helpful information gathering and design tools because they are easy to create, iterate, and read. If kept in public spaces, they can become an accessible source of insights into user experiences (either as they exist today or as they are envisioned to become). Unlike prose documents or complex flowcharts, Maps make it easy to quickly extract and understand the end-to-end user experience and/or focus on details of interest.

Design Maps can help you test new experiences before you build a new product. Moreover, Design Maps will help you translate what you know about your personas into designs for new experiences that your product could support. Once you design the experience you want to create, it is relatively easy to create features to support that experience.

What Are Design Maps?

Design Maps tell stories that look into the future; these stories describe how your personas will behave once your new product is built. Those familiar with scenario-based design will recognize that Design Maps have a distinct similarity to scenarios (see Carroll, 1995). Scenarios are short prose stories that describe how aspects of your product will be—or should be—used. Design Maps are both a special type of scenario and a process by which to create scenarios and modify them. Design Maps are flow-chart version of many scenarios "strung together" to create a big picture of the experience your product will support. Design Maps are inexpensive (both in terms of time and materials) and they are most helpful when built before paper prototypes, and certainly before any code is written (see Fig. 51.8).

Which Processes Should I Design Map?

Design Maps depict end-to-end experiences (not specific features or widgets). You can create Design Maps to explore any experience you want to create as it relates to any aspect of your new product's design. The Design Maps you create should explore the ways that your personas achieve the goals that you have established for them. Remember that their roles and goals may change in your new designs.

Design Map for the "big picture." A "big picture" Map shows the entire experience end-to-end and therefore describes activities in very broad terms. Think of this "overview" Design Map as analogous to a map of the United States with a line drawn on it to show the route of a cross-country driving trip; the overview Map should give the reader a general sense of direction and the order of progression, but should not contain details.

Design Maps for achieving major milestones. "Major milestone" Maps should "fit into" the overall Map, but should explore individual goals and tasks more specifically. In the cross-country trip example, a "major milestone" Map would be the equivalent of a highway map showing the roads you used to traverse a single state.

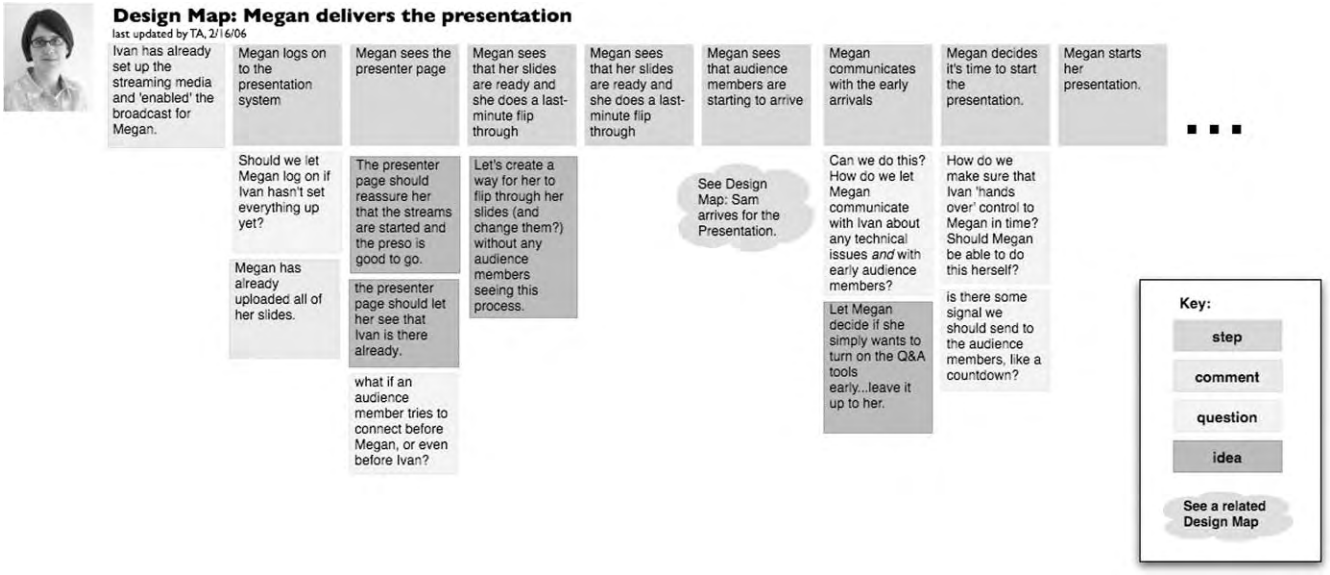


FIGURE 51.8. A Design Map exploring how Megan starts an online presentation. Design Maps are created by your team and explore the new experiences you are going to build into your product.

Design Maps for critical details. "Critical details" Maps should fit into the milestones Maps much as the milestones fit into the big picture. These Maps explore very specific details of particular tasks, the way an enlargement of a downtown area shows the specific details of the ways streets crisscross a city.

How to Create a Design Map

Maps show steps in a process or experience sequentially, with any questions, comments, or ideas regarding these steps arrayed underneath. Finished Design Maps are large sheets of paper covered with color-coded Post-It™ (or "sticky") notes that describe the user experiences related to your product. Design Maps are created by the design team without the participation of users.

The elements of a Map: titles, steps, questions, comments, and design ideas. All Maps have five basic building blocks: (a) titles, (b) steps, (c) questions, (d) comments, and (e) design ideas. Titles should identify the goals or tasks the Map depicts. Steps should be arrayed horizontally, with related comments, questions, and design ideas arranged under the steps they reference. You can read across the row of steps to get a sense of the process from end to end (e.g., the steps in a task taken to reach a goal), or you can focus on a subset of the steps and read down the columns to understand related questions and ideas (see Fig. 51.9).

Steps (blue sticky notes): These are the "verbs" or the "backbone" of the process. The facilitator of the Mapping exercise places steps horizontally across the Map. A good way to elicit steps is to ask, "What will the persona do next?" Steps are the building blocks of tasks.

Comments (green sticky notes): Comments are qualifying statements about steps. They are the most flexible elements on a Map. Comments can describe behaviors, habits, awareness or lack of awareness of features or alternative actions, or even qualities of objects. If you hear an important piece of information, but it is not a step, question, or design idea, record it as a green comment.

For example, in our Megan Delivers the Presentation Map, the comment "Megan has already uploaded all of her slides" is a note about her actions that could be significant with respect to the rest of the experience. The comment is not a step, but it relates to the step listed above: "Megan logs on to the presentation system." The comment, in this example, serves to remind the facilitator and Map readers that we are assuming that the slides are uploaded in a separate, previous series of actions in this particular Design Map. If this comment were to change (in this example, if we were to change or design the process so that the slides are uploaded immediately before the presentation), it could affect all of the steps, comments, questions, and ideas in the remainder of the Map.

Questions (yellow sticky notes): Yellow "questions" are the most useful interview management tool of the Mapper. When you first start Mapping any process, you will identify many questions, some indicating areas where you need clarification and some that express your Mapping participants' issues. In fact, you will probably encounter so many questions that the sheer volume and importance of them will threaten to derail your attempt to Map the entire process. Listing the questions on the Map allows you to record and move past them quickly so that you can capture as much of the process as possible without

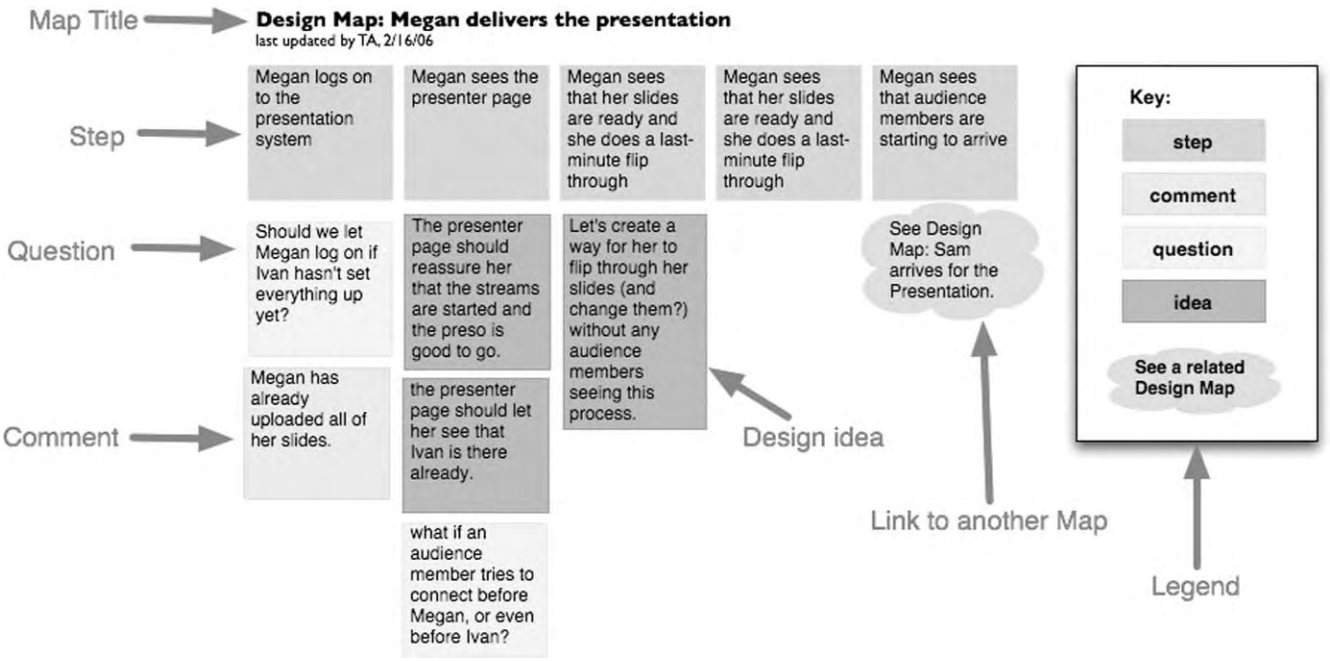


FIGURE 51.9. Title, steps, questions, comments, and design ideas in a Design Map.

being derailed. Once you create a Map that captures most of an end-to-end process, you can loop back and track down answers to the questions you have identified.

Create a yellow question when

- You have questions about the process.
- You are not quite sure what will, should, or could happen next.
- Anyone participating in the Mapping session begins to belabor a point.

Design Ideas (pink sticky notes): As you create your Map, you will inevitably think of, or be presented with, an assortment of ideas for specific features, widgets, or even things like marketing messages. Your Mapping goal is to create a solid picture of the end-to-end experience you want to create for your persona. While you do not want to allow your Mapping session to turn into a discussion of specific new features, you also should not discard good ideas just because they come up at the "wrong time." For example, in Fig. 51.8, one pink sticky note says, "Let's create a way for her to flip through her slides (and change them?) without any audience members seeing this process." This is an interesting idea and worth capturing. Ask anyone who comes up with a design idea to record it on a pink sticky note, place it on the Map, and move on.

Facilitating a Design Mapping session. Encourage Mapping participants to focus on the *experience*, not on the *tool*. The goal is not to have a Map that tells you, "The serial number registration tracking database will feed the score records to the page via ASP," but one that says, "Megan can see that Ivan is already logged on."

During Design Mapping sessions, remind your team to consider the following questions:

- Do the tasks assigned to personas in the Design Map correspond to your personas' skills? For example, if Sam the Sales Rep is expected to spend an hour doing technical preparation so that he can attend the presentation, he is not being well served by the new design. If Megan's Design Map allows her to answer questions by presenting new content on the fly, you are on the right track.
- Does the new process being constructed in the Map offer undeniable advantages to the personas over the old way of doing things?
- Are we assuming things have to be done a certain way just because it is the way they are done now?

As you move through a Mapping session, remember to table questions that might sidetrack your work by providing everyone with yellow sticky notes and encouraging participants to write down difficult questions and issues and post them on the map. During your Design Mapping session, you might hear a comment such as, "Well, if we're assuming Sam can access the Internet using a broadband connection from wherever he is, even if he's on the road, we can assume he won't have any trouble viewing the presentation. It would also be great if we could assume he has all the media players installed, and the latest version of the browsers. This makes things easier for us." This is a good opportunity to refer to your personas. Do you have information about Sam's technical setup? How likely is it that his computer will have the latest media players? Is this likely to be something that is done for him, or something he has to figure out for himself? Your personas will be able to immediately answer some of these questions; other questions will have to go onto to the Map to be answered later. In this particular example, you might decide to create another Map that assumes that Sam does not have everything he needs loaded on his machine. What process are you going to design to make it easy for him to prepare for and attend a presentation in this case?

Sometimes you will want to move quickly, placing blue steps across the top of the Map and filling in details later. Other times your team might find it most effective to hash out the details under each step before moving to the next one. In either case, you will want to limit Mapping sessions to two to three hours each.

After each session, follow up on any questions or issues raised and add answers to the Map. It can be useful to convert the sticky notes paper versions of your Maps into electronic versions in Microsoft Visio or a similar tool. This makes it easy to for participants to review progress and quickly scan for new material. The electronic versions are useful for printing in various formats and sending to stakeholders for review at a distance.

Use Design Maps to Create Wireframes

Wireframes tend to evolve naturally from Design Maps. Once you and your team have agreed on the experience you want to facilitate for your personas, it is relatively easy to use the steps, assumptions, questions, and design ideas in the Maps to create wireframes of the product's UI.

With your team, identify the columns of the Design Map that "go together" and should be grouped on a single interface (see Fig. 51.10). At this point, the user interface designer or graphic designer should be heavily involved. Consider what information you are collecting from the personas and when you are collecting it so that you can plan to display it on the UI at the right times (e.g., if you have not yet asked the persona for his or her name yet, you cannot create a wireframe for a personalized interface).

The Benefits of Design Maps

Perhaps the most important benefit of Design Maps (and wireframes) is that they hold and communicate a shared vision of the project for your entire team. Seeing the "big picture" early on in the project helps motivate everyone toward the common goal of making it real. Design Maps enable your entire team to "see" the product from the personas' points of view, giving the architects of the product the opportunity to understand and empathize with users. A deeper understanding of the planned product and personas early on in the development process can enhance each team member's work on your product, whether they are coding, marketing, managing, testing, funding or selling it.

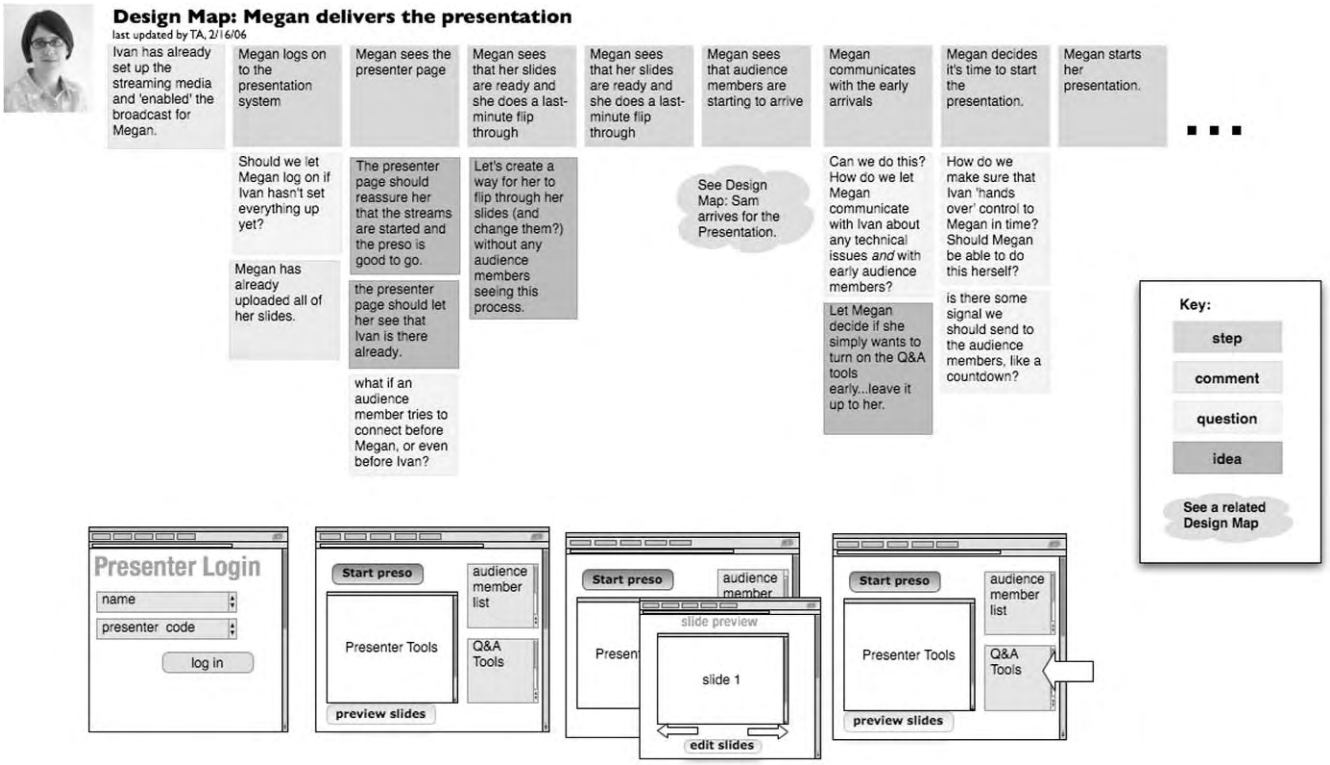


FIGURE 51.10. Wireframes evolve naturally from Design Maps. You and your Mapping team can evaluate which process steps belong together and design the interface flow accordingly.

Design Maps (and their associated wireframes) are a perfect document to work from when communicating project plans to stakeholders. Once your Design Maps are completed, you can use them to perform design walk-throughs of your product with other team members. Have someone read through the Map and another person check the prototype or the product to make sure the Mapped process is supported by the product's design.

As your development team begins coding your new product, check in frequently to evaluate what they are building against what you *thought* they were going to build. Use the Design Maps to "test" the emerging product: does the product being built actually support the experience you have designed? The Design Maps you created define a task flow; when the product changes during development, use the Maps to evaluate whether or not the changes support the tasks you identified.

Design Mapping Illuminates—and Fixes!—Holes in Other Kinds of Specification Documents

Raina Brody & Sylvia Olveda,
Usability Specialists, Amazon.com

The Amazon usability team works with many different project teams from all over the company. For one of our projects, we were working with a team that was designing and developing a new feature for the website. The team had already created three primary personas and a very large document full of use cases, which specified many features and user interface elements. The use cases and user interface specifications all referred to their personas by name. In short, it looked like they had done a lot of good, user-centered work and were pretty far along the path toward creating a good solution. We asked the designer and the design manager why they felt they needed our help.

They replied, "All of this stuff looks really good, but there are enormous holes in the use cases when it comes to the actual UI, the experience flow, and the technical implementation/specifications. Some of the technical services and features these use cases call for don't exist and simply can't be built—and the rest of the team doesn't seem to get this." We decided to lead a few Design Mapping sessions to help them identify and document the holes and to communicate their concerns to the rest of the team.

We set up a meeting with the project team (including members of the business team and design and development teams) and asked them to pick a couple of the most important use

cases. They were a bit hesistant at first, because some members felt that they had already done all of this work in the use case document. After describing their primary use cases, we started the Design Mapping exercise. Very soon, it became clear that there were holes in the existing documentation. For example, some of the original use cases assumed the existence of data types and search services that simply did not exist, and that were not scoped as part of the project schedule. The Design Map helped us discover that very important elements in their use cases had not yet been thought about.

The Design Mapping process really started going well when the project team members realized that we, the usability team, were not trying to design their product for them. Instead, we were serving as facilitators to help them come up with and discuss their ideas from a customer experience and technical feasibility perspective. Of course, it was our job to ask (and sometimes answer) some difficult questions about whether customers would be able to discover and/or understand some of the user interface elements the team came up with.

The Mapping process generated excitement from the project team and some of the best ideas came from developers. After our first meetings, the Mapping sessions went well and went very fast. The team agreed that the Design Mapping process helped them identify what was hard, what was easy, and what they could really do. The team also realized that while the original use case document contained a lot of good ideas, the complexities of the actual implementation were not made as apparent as they were with the Mapping session.

At the end of the project, we asked the team how the Design Mapping and personas helped with the project. The answers we got were very encouraging:

- From a key developer: "Design mapping helped everyone on the team to really focus on the project from our customer's point-of-view. It took a lot of disparate opinions and ideas about what we wanted to build and provided a common focus. It was responsible for some very productive brainstorming."
- From the Technical Program Manager: "The personas and Mapping process are invaluable tools designing a product that is customer-centric; together they have the power to overcome designs driven by emotion and bias: two things commonly found in every product design cycle."
- From the Product Manager: "The Design Mapping we did at the beginning of our project has proven to be an invaluable tool to keep the team focused on what is best for our target customers. It was also an extremely helpful team-building exercise."
- From the Team Manager: "The design mapping process helps us develop the right features for the right customers. Project teams are hesitant at first because it seems like a significant time investment, but once they go through the process, they find it invaluable. It actually saves time in the long run by making sure teams make better decisions up front."

We learned some valuable lessons from this Mapping exercise:

- If you are Mapping with a team that has never tried it before, do not start by Mapping a controversial feature or experience. Instead, start with one of the features or experiences that seems well defined and clear. This will help them team understand the Mapping process and will show them that you do not intend to take over the design of their product.
- Encourage teams to do use cases *after* they do Design Maps. Design Maps generate lots of use cases, and they have the added benefit of clearly illustrating how use cases fit together in an end–to-end customer or user experience.
- Present Design Mapping as a technique to facilitate the product team's design process. It is important that the team still feel in control, even in a situation where you are leading a Mapping exercise.
- Create a Mapping plan after your first Mapping meeting. After you do one Map, it will be easy to identify other user experiences you want to Map. Revisit your Mapping schedule after you do the first two or three Maps to integrate new ideas for Maps and a new perspective on how long the Mapping process will take.
- Think about who you want to invite and who you need to invite to Mapping sessions. We always facilitated, and invited at least one designer, the technical program or project manager, between one and three members of the technical/development team, and the product manager.

Word is spreading fast, and now we have so many requests to facilitate Design Mapping sessions we can hardly keep up with them.

Phase 5: Persona Lifetime Achievement & Retirement

Once the project or product is completed, it is time to think about what has been accomplished, and also to prepare for the next project. You will want to assess how effective the persona method was for your team and product development process. If you are beginning to think about the next product (or next version of the product you have just released) you will need to decide whether, and how, you will reuse your existing personas and the information they contain.

The end of a product cycle is a good time to assess the effectiveness of personas for the team and to take stock of lessons learned for the next time. How did the development team accept the method? Were your personas useful? To what extent were they accurate and precise? We will provide suggestions and tools you can use to validate the use of personas in the development process and to determine if the persona effort was worth the effort and resources it required. Did personas change the product? Did they change your design and development process? User-centered designers are constantly under pressure

to validate the worth and ROI of their activities and personas can be useful tools for measuring the success of both the product and of the UCD activities as a whole.

Recall the questions we recommended you answer in Phase 1, Family Planning, related to "organizational introspection." Assessing return on investment is, in simple terms, working to answer those original questions (listed in the left column of Fig. 51.11, below) in terms that are meaningful now that you are done with the effort (e.g., those questions listed in the right column of Fig. 51.11).

Retirement

Depending on the nature of your products, you might be able to reuse the personas or "reincarnate" some of your persona data in new personas. That is, you will need to decide what to do with your "old" personas as you prepare for your next project. Do your personas retire? Do they change over time? Do they buy your product and start using it? Can other product teams utilize your personas or some portion of the information in them (e.g., are they reincarnated)?

In most cases, you will decide to use some combination of direct reuse (using them again without alteration), reincarnation (reusing some their content and related data) or retirement (discarding or completely replacing some of the personas). Note that, if your persona effort has been a success, retirement and reincarnation can be a bit tricky; to have "room" for the next set of personas, you will need to help your organization let go of the personas they have come to know so well.

Before you decide what to do with your personas, you need to revisit the data sources you used to create them. If you are about to start work on the next version of the product you have just released, it is likely that many (but probably not all) of your data sources are still relevant and you can reuse entire personas or some of the information in the personas. If you're moving on to create a completely different product or if there have been major shifts in strategy, then perhaps only a few of the data sources may still be relevant (e.g., those that relate to your company or to the general product space in which you work).

Questions You Should Ask During Family Planning	**Questions You Will Ask During Lifetime Achievement**
What resources do we have for personas and other UCD activities?	How much did the persona effort actually cost?
What product problems do we want to solve with personas?	Has the product improved? How much, and in what ways?
What process problems do we want to solve with personas?	Has the process improved? In what ways?
How can we ensure that the personas will be accepted and used by our colleagues?	Were personas perceived as helpful? Has the company's focus on users improved? In what ways?

FIGURE 51.11. The work you did during Family planning is highly related to what you will need to do in Lifetime Achievement.

Reusing Your Personas

If you're building a new version of your product, or a new product for the same audience, you might find that many of your personas can be reused. Your personas could be reused by the same team that used them originally, by a new design and development team, or perhaps by a team in some other part of your company, like marketing, sales, or product support.

When you created your personas, you (a) assimilated your data, (b) created persona skeletons, (c) prioritized the skeletons, and (d) built some or all of the skeletons into full personas. When you move on to the next version of your product, you can reevaluate the primary versus secondary classification for each of your original personas. You might decide to demote one of your primary personas and promote one of your secondary personas; this promotion/demotion is especially useful if you are building a new version of the same project but your company has decided to focus on a slightly different user base. Additionally, you can revisit some of the persona skeletons you created but never developed for the first project; it is possible that one of these sketches would just right for the new project. If so, you have a tremendous head start and can simply build up the sketch into a full persona.

Reincarnating Personas

If some of the data in your personas is still relevant, but the personas you originally created are not, you can create "reincarnated" personas by reusing "old" data in new personas. If the products you develop serve users in a specific market segment or industry, you will find many data sources that stay relevant no matter what project you are working on. If the data sources are still relevant, but the particular personas you have created are not, it is important to do some research and find some additional sources.

Once you have collected the appropriate set of data (which will include sources you've already used), you can revisit the processes described in the Conception & Gestation section and reassimilate data points according to the issues you're finding related to your new project.

Retiring Your Personas

You might decide that you do not want to reuse or reincarnate your personas or their underlying data at all. There are many reasons to retire personas before moving on to the next project:

- The current project is significantly different from the last project.
- The users' goals have changed.
- Your company adopted a new strategy or is targeting a different user base.

- There are significant changes to the environment in which your product will be used, such as new technologies or new competitive products that have "changed the landscape (e.g., the advent of streaming media and broadband in the home, or Bluetooth technology for computing).

When you determine that a persona or set of personas is no longer relevant, it is a good idea to *officially* retire the persona(s) before moving on.

Why "officially"? Why not just take down the posters and start working on a new set of personas? Because if you have done your job well, you have made the personas incredibly memorable, and all your work has paid off; people in your organization have absorbed various amounts of information about the primary personas and are accustomed to thinking about them. If you try to introduce a new set of personas "on top of" an old set, you run the risk of confusing your team, which will destroy the clarity that personas are supposed to provide.

You can't reach into your colleagues' heads and erase everything they know about the personas you've been using, but you have to find a way to help them move past the old personas and let go of the (no longer relevant) information they contain. This can be as simple as an e-mail announcement that the old set of personas is retiring and why. If you are moving on to a totally different product or a new strategy, most of your colleagues would know about the switch and the retirement announcement will make sense to them. Use this as an opportunity to invite feedback from the team on the ways personas helped or didn't help them do their jobs; you can use this feedback to tweak your customized persona lifecycle the next time around.

Moving on to the Next Persona Effort

Finally, it is worth noting that the activities we recommend at the end of one persona project function to prepare you for the next persona project; regaining control over the personas and evaluating the success of the effort will help you be even more successful in your next effort. The Lifetime Achievement, Reuse, & Retirement phase provides an excellent opportunity to touch base with your core team members and with other stakeholders to talk about how things went. As you dive back into Family Planning, you will want to predict the new issues you may encounter. This final lifecycle phase is a great time to have a postmortem to talk about what improved, stayed the same, or worsened during or due to your persona effort.

SUMMARY

Personas can be an invaluable asset to product design and development teams, but they must be created and used with care. The best advice we can give you as you embark on your own persona efforts is to keep in mind the following axioms that are at the heart of the persona lifecycle approach:

- Building personas from assumptions is good; building personas from data is much, much better.
- Personas are a highly memorable, inherently usable communication tool *if* they are communicated well.
- Personas can be initiated by executives or first used as part of a bottom-up, grass roots experiment, but eventually need support at all levels of an organization.
- Personas are not a stand-alone user-centered design process, but should be integrated into existing processes and used to augment existing tools.
- Personas can help bring a user-centered focus into even a stubbornly technology-focused organization.
- Effective persona efforts require organizational introspection and strategic thinking.
- Personas *can* be created fast and show their value quickly, but if you want to get the full value of personas, you will have to commit to a significant investment of time and resources.

In short, "the devil is in the details" when it comes to launching a persona effort within an organization. Perhaps this is the reason that persona efforts are inevitably more work than you think they are going to be. If you can dedicate some time, and can get some help from at least one colleague, the persona lifecycle will enable you to integrate persona creation and use methods into your existing processes. Equally importantly, it will provide you with data-driven, user-focused reasons to change aspects of existing processes that are not working well for team and are resulting in problematic products.

Personas in the Present—And Looking Toward the Future

Kim Goodwin, Vice President and General Manager, Cooper

As for the future of personas, I have heard a number of people say that personas are just the latest fad, and no one will care about them in a few years. It is true that some people are using things they call "personas" just because they are the latest craze, and no doubt, the popularity of personas will fade quickly with those people who find that a photo, a name, and a collection of personal tidbits do not accomplish much. People who look to personas to solve all of their problems will also be disappointed; however, those practitioners who take the time to master personas—which are easy to understand but not so easy to develop and use—will find that personas are one tool they will never put down.

In the last few years, Cooper designers have been using personas for more than product definition and interaction design guidance. We find that experience goals (how a persona wants to feel, as opposed to the end goals we usually focus on with personas) are invaluable in guiding the aesthetic aspects of the visual, auditory, and physical design. Does your online banking persona want to feel her money is secure? A lime green background and Comic Sans typeface are probably poor choices. Does a critically ill person want to be reminded of his condition? No, so perhaps his home

healthcare device should not look as if it belongs in a hospital. We have used personas for designing business processes, organizations, and training curricula as well.

Personas have been adopted by certain parts of companies, but not yet by others. In many cases, we see companies that use personas for one product but not others, or who rely heavily on personas to design their websites, but do not yet use them to inform the design of other customer-facing systems. I believe the lack of adoption is not due to a failing of the method, but to a common failing of organizations: silos. The website or product design teams are generally isolated from other parts of a company, so those other groups are unaware of the advances those teams are making. In addition, there is seldom anyone with overall responsibility for a customer's or user's experience with that company. To realize the true potential of their investment in personas, companies will need to look for other opportunities to use them, even outside of electronic systems. For example, a couple of years ago, I spoke with a human resources publication interested in applying employee personas to benefits plans.

There is no question in my mind: personas are here to stay, and we will see them showing up in more and more places. After all, personas are about understanding our fellow human beings, and that will never go out of style.

Areas for Further Investigation

While we are obviously fans of the persona approach, personas are not proven. Indeed, our experiences with them have proven to us that personas have their limits; personas are not a panacea, and they are not right for every situation or organization. The details of these shortcomings are not well understood at this time, though some research does exist (see, e.g., Blomquist & Arvola, 2002; Rönkkö, Hellman, Kilander, & Dittrich, 2004). Clearly, personas do not help a development team build better products merely by their presence. As we argue here (and in our book, *The Persona Lifecycle* [Pruitt & Adlin, 2006]), personas must be assimilated by the product team and explicitly used in design and development activities. Nevertheless, what uses of personas have the most value? What characteristics of personas are the most critical? We have provided some process and guidance to help practitioners do personas successfully, but at present, this is based on practice and experience, not science. We need several rigorous evaluations of the method to better understand its benefits and limitations.

The process we recommend for persona creation, while doable and generally effective, is not particularly precise. Given the same data as input, different groups of people will likely create somewhat different personas. Sinha (2003) and others (including us) have proposed quantitative analysis methods as the basis for persona creation to enhance repeatability and accuracy. While this is certainly a plausible move in the right direction, it is the application of qualitative data that helps make a persona into a persona. We do not believe it solves the problem. Further, the notion that personas can be abstracted into elemental characteristics or directly reused across multiple product lines is contentious. More investigation is needed here.

Finally, there are some cases where development teams have attempted to create representations of things other than people (e.g., organizational archetypes—a persona of a company) to be used, like personas, in the development of products or services. We believe that such representations, while potentially useful, do not have the power and impact that representations of people have. They are not as memorable or provocative as personas—we relate to personas in ways that we cannot relate to other abstractions. We suggest this as an interesting line of research.

References

Blomquist, Å., & Arvola, M. (2002). Personas in action: Ethnography in an interaction design team. *Proceedings of the Second Nordic Conference on Human-Computer Interaction, NordiCHI* (pp. 197–200). New York: ACM Press.

Carroll, J. (Ed.) (1995). *Scenario-based design: Envisioning work and technology in system development*. New York: Wiley.

Carroll, J. (2000a). Five reasons for scenario-based design. *Interacting with Computers, 13*(1), 43–60.

Carroll, J. (2000b). *Making use: Scenario-based design of human-computer interactions.* Cambridge, MA: MIT Press.

Constantine, L., & Lockwood, L. (2001, August). Personas. *forUse, 15*. Retrieved March 21, 2007, from http://www.foruse.com/newsletter/foruse15.htm

Constantine, L., & Lockwood, L. (2002, October). Modeling: Persona popularity and role relationships. *forUse, 26*. Retrieved from http://www.foruse.com/newsletter/foruse26.htm

Cooper, A. (1999). *The inmates are running the asylum.* Indianapolis, IN: Macmillan.

Cooper, A., & Reimann, R. (2003). *About Face 2.0: The essentials of interaction design.* Indianapolis: Wiley Publishing.

Dreyfuss, H. (2003). *Designing for people*. New York: Allworth Press. (Original work published in 1955)

Freed, J. (2004). Ahead of the game: Best Buy revamps its stores to be ready for the challenge from a new line of electronics retailers. Associated Press. Retrieved March 21, 2007 from http://www.projo.com/business/content/projo_20040520_best20x.201cc9.html

Grudin, J. (1990). Constraints in product development organizations. *Proceedings of the Participatory Design Conference* Palo Alto, CA: Computer Professional for Social Responsibility (pp. 14–21).

Grudin, J. (1993). Obstacles to participatory design in large product development organizations. In D. Schuler & A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 99–119). Mahwah, NJ: Lawrence Erlbaum Associates.

Grudin, J., & Pruitt, J. (2002). Personas, participatory design, and product development: An infrastructure for engagement. *Proceedings of the PDC 2002* (pp. 144–161). New York: ACM Press.

Hourihan, M. (2002, March). Taking the "you" out of user: My experience using Personas. *Boxes and Arrows*. Retrieved February 20, 2006 from http://boxesandarrows.com/archives/002330.php

Jacobson, I. (1995). The use-case construct in object-oriented software engineering. In J. M. Carroll (Ed.), *Scenario-based design.* New York: Wiley.

Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (1992). *Object-oriented software engineering: A use case driven approach.* Reading, MA: Addison-Wesley.

Levinson, M. (2003). Website redesign: How to play to your audience. Retrieved March 21, 2007 from http://www.cio.com/archive/111503/play.html

McGovern, Gerry. (2002). *Don't call people users*. gerrymcgovern.com.

Mello, S. (2003). *Customer-centric product definition: The key to great product development.* Boston: PDC Professional Publishing.

Moore, G. A. (1991). *Crossing the chasm: Marketing and selling high-tech products to mainstream customers.* New York: Harper Collins Publishers. (Revised in 2002)

Nielsen, J. (1993). *Usability engineering.* Boston: Academic Press.

Pruitt, J., & Adlin, T. (2006). *The persona lifecycle: Keeping people in mind throughout product design.* San Francisco, CA: Morgan Kaufmann Press.

Rönkkö, K., Hellman, M., Kilander, B., & Dittrich, Y. (2004). Personas is not applicable: Local remedies interpreted in a wider context. Paper presented at the Participatory Design Conference, Toronto, Canada.

Sinha, R. (2003). Persona development for information-rich domains. *Proceedings of CHI 2003* (pp. 830–831). New York: ACM Press.

Sissors, J. (1966). What is a market. *Journal of Marketing, 30*, 17–21.

Star, S. L. & J. R. Griesemer. (1989). Institutional ecology, 'translations,' and boundary objects: Amateurs and professionals in Berkeley's museum of vertebrate zoology, 1907–1939. Reprinted in Mario Biagioli (ed.), *The science studies reader* (pp. 505–524). New York: Routledge.

Upshaw, L. (1995). *Building brand identity: A strategy for success in a hostile marketplace.* New York: John Wiley & Sons.

Weinstein, A. (1998). *Defining your market: Winning strategies for high-tech, industrial, and service firms.* New York: Haworth Press.

•52•

PROTOTYPING TOOLS AND TECHNIQUES

Michel Beaudouin-Lafon
Université Paris-Sud

Wendy E. Mackay
INRIA

Introduction **1018**
What Is a Prototype? 1018
Prototypes as Design Artifacts **1018**
Representation 1018
Precision 1019
Interactivity 1019
Evolution 1020
Prototypes and the Design Process **1020**
User-Centered Design 1020
Participatory Design 1021
Exploring the Design Space 1021
Expanding the Design Space: Generating Ideas 1022
Contracting the Design Space:
Selecting Alternatives 1023
Prototyping Strategies 1024
Horizontal prototypes 1024
Vertical prototypes 1024
Task-oriented prototypes 1024
Scenario-based prototypes 1024
Rapid Prototyping **1025**
Offline Rapid Prototyping Techniques 1025
Paper and pencil 1025
Mock ups 1025
Wizard of Oz 1026
Video prototyping 1026
Online Rapid Prototyping Techniques 1028
Noninteractive simulations 1029
Interactive simulations 1030
Scripting languages 1030
Iterative and Evolutionary Prototypes **1032**
User Interface Toolkits 1033
User Interface Builders 1033
User Interface Development Environments 1034
Prototyping Mixed Reality and
Pervasive Computing Systems **1036**
Conclusion **1037**
References **1038**

INTRODUCTION

"A good design is better than you think."
(Heftman, as cited in Raskin, 2000).

Design is about making choices. In many fields that require creativity and engineering skill, such as architecture and automobile design, prototypes both inform the design process and help designers select the best solution.

This chapter describes tools and techniques for using prototypes to design interactive systems. The goal is to illustrate how they can help designers generate and share new ideas, get feedback from users or customers, choose among design alternatives and articulate reasons for their final choices.

We begin with our definition of a prototype and then discuss prototypes as design artifacts, introducing four dimensions for analyzing them. We then discuss the role of prototyping within the design process, in particular the concept of a design space and of how it is expanded and contracted by generating and selecting design ideas. The following sections describe specific prototyping approaches: rapid prototyping, both offline and online, for early stages of design, and iterative and evolutionary prototyping, which use online development tools. Finally, we address the specific issue of prototyping mixed reality and pervasive computing systems.

What Is a Prototype?

We define a *prototype* as a concrete representation of part or all of an interactive system. A prototype is a tangible artifact, not an abstract description that requires interpretation. Designers, as well as managers, developers, customers and end users, can use these artifacts to envision and to reflect upon the final system.

Note that other fields may define prototype differently. For example, an architectural prototype is a scaled-down model of the final building. This is not possible for prototypes of interactive systems: The designer may limit the amount of information the prototype can handle, but the actual user interface must be presented at full scale. Thus, a prototype interface to a database may handle only a small subset of the final database but must still present a full-size display and interaction techniques. Fashion designers create another type of prototype, a full-scale, one-of-a-kind model, such as a handmade dress sample. Although in haute couture, this prototype may also be the final product, the ready-to-wear market requires additional design phases to create a design that can be mass-produced in a range of sizes. Some interactive system prototypes begin as one-of-a-kind models that are then distributed widely (since the cost of duplicating software is so low). However, most successful software prototypes evolve into the final product and then continue to evolve as new versions of the software are released.

Hardware and software engineers often create prototypes to study the feasibility of a technical process. They conduct systematic, scientific evaluations with respect to predefined benchmarks and, by systematically varying parameters, fine tune the system. Designers in creative fields, such as typography or graphic design, create prototypes to express ideas and reflect on them. This approach is intuitive, oriented more to discovery and generation of new ideas than to evaluation of existing ideas.

HCI is a multidisciplinary field that combines elements of science, engineering, and design (Mackay & Fayard, 1997; Dijkstra-Erikson et al., 2001). Prototyping is primarily a design activity, although we use software engineering to ensure that software prototypes evolve into technically sound working systems and we use scientific methods to study the effectiveness of particular designs.

PROTOTYPES AS DESIGN ARTIFACTS

We can look at prototypes as both concrete artifacts in their own right or as important components of the design process. When viewed as artifacts, successful prototypes have several characteristics: They support creativity, helping the developer to capture and generate ideas, facilitate the exploration of a design space, and uncover relevant information about users and their work practices. They encourage communication, helping designers, engineers, managers, software developers, customers and users to discuss options and interact with each other. They also permit early evaluation since they can be tested in various ways, including traditional usability studies and informal user feedback, throughout the design process.

We can analyze prototypes and prototyping techniques along four dimensions:

- *Representation* describes the form of the prototype, such as sets of paper sketches or computer simulations;
- *Precision* describes the level of detail at which the prototype is to be evaluated, such as informal and rough or highly polished;
- *Interactivity* describes the extent to which the user can actually interact with the prototype, such as "watch only" or fully interactive;
- *Evolution* describes the expected life cycle of the prototype, such as throwaway or iterative.

Representation

Prototypes serve different purposes and thus take different forms. A series of quick sketches on paper can be considered a prototype; so can a detailed computer simulation. Both are useful; both help the designer in different ways. We distinguish between two basic forms of representation: offline and online.

Offline prototypes (also called "paper prototypes") do not require a computer. They include paper sketches, illustrated storyboards, cardboard mock ups and videos. The most salient characteristics of offline prototypes (of interactive systems) is that they are created quickly, usually in the early stages of design, and they are usually thrown away when they have served their purposes.

Online prototypes (also called "software prototypes") run on a computer. They include computer animations, interactive video presentations, programs written with scripting languages, and applications developed with interface builders. The cost of

producing online prototypes is usually higher, and may require skilled programmers to implement advanced interaction and visualization techniques or to meet tight performance constraints. Software prototypes are usually more effective in the later stages of design, when the basic design strategy has been decided.

In our experience, programmers often argue in favor of software prototypes even at the earliest stages of design. Because they are already familiar with a programming language, these programmers believe it will be faster and more useful to write code than to "waste time" creating paper prototypes. In 20 years of prototyping, in both research and industrial settings, we have yet to find a situation in which this is true.

First, offline prototypes are very inexpensive and quick. These permit a very rapid iteration cycle and help prevent the designer from becoming overly attached to the first possible solution. Offline prototypes make it easier to explore the design space, examining a variety of design alternatives and choosing the most effective solution. Online prototypes introduce an intermediary between the idea and the implementation, slowing down the design cycle.

Second, offline prototypes are less likely to constrain how the designer thinks. Every programming language or development environment imposes constraints on the interface, limiting creativity and restricting the number of ideas considered. If a particular tool makes it easy to create scroll-bars and pull-down menus and difficult to create a zoomable interface, the designer is likely to limit the interface accordingly. Considering a wider range of alternatives, even if the developer ends up using a standard set of interface widgets, usually results in a more effective design.

Finally, and perhaps most importantly, offline prototypes can be created by a wide range of people, not just programmers. Thus all types of designers, technical or otherwise, as well as users, managers, and other interested parties, can all contribute on an equal basis. Unlike programming software, modifying a storyboard or cardboard mock up requires no particular technical skill. Collaborating on paper prototypes not only increases participation in the design process, but also improves communication among team members and increases the likelihood that the final design solution will be well accepted.

Although we believe strongly in offline prototypes, they are not a panacea. In some situations, they are insufficient to evaluate fully a particular design idea. For example, interfaces requiring rapid feedback to users or complex, dynamic visualizations usually require software prototypes. However, particularly when using video and Wizard of Oz techniques, offline prototypes can be used to create very sophisticated representations of the system.

Prototyping is an iterative process and all prototypes provide information about some aspects while ignoring others. The designer must consider the purpose of the prototype (Houde & Hill, 1997) at each stage of the design process and choose the representation that is best suited to the current design question.

Precision

Prototypes are explicit representations that help designers, engineers and users reason about the system being built. By their nature, prototypes require details. A verbal description such as "the user opens the file" or "the system displays the results" provides no information about what the user actually does or sees. Prototypes force designers to show the interaction: just how does the user open the file and what are the specific results that appear on the screen?

Precision refers to the relevance of details with respect to the purpose of the prototype.[1] For example, when sketching a dialog box, the designer specifies its size, the positions of each field and the titles of each label. However not all these details are relevant to the goal of the prototype: it may be necessary to show where the labels are, but too early to choose the text. The designer can convey this by writing nonsense words or drawing squiggles, which shows the need for labels without specifying their actual content.

Although it may seem contradictory, a detailed representation need not be precise. This is an important characteristic of prototypes: those parts of the prototype that are not precise are those open for future discussion or for exploration of the design space. Yet they need to be incarnated in some form so the prototype can be evaluated and iterated.

The level of precision usually increases as successive prototypes are developed and more and more details are set. The forms of the prototypes reflect their level of precision: sketches tend not to be precise, whereas computer simulations are usually very precise. Graphic designers often prefer using hand sketches for early prototypes because the drawing style can directly reflect what is precise and what is not: the wiggly shape of an object or a squiggle that represents a label are directly perceived as imprecise. This is more difficult to achieve with an online drawing tool or a user interface builder.

The form of the prototype must be adapted to the desired level of precision. Precision defines the tension between what the prototype states (relevant details) and what the prototype leaves open (irrelevant details). What the prototype states is subject to evaluation; what the prototype leaves open is subject to more discussion and design space exploration.

Interactivity

An important characteristic of HCI systems is that they are interactive: users both respond to them and act upon them. Unfortunately, designing effective interaction is difficult: many interactive systems (including many websites) have a good look but a poor feel. HCI designers can draw from a long tradition in visual design for the former, but have relatively little experience with how interactive software systems should be used: personal computers have only been commonplace for a couple decades. Another problem is that the quality of interaction is tightly linked to the end users and a deep understanding of their work prac-

[1]Note that the terms *low-fidelity* and *high-fidelity* prototypes are often used in the literature. We prefer the term *precision* because it refers to the content of the prototype itself, not its relationship to the final, as-yet-undefined system.

tices: a word processor designed for a professional typographer requires a different interaction design than one designed for secretaries, even though ostensibly they serve similar purposes. Designers must take the context of use into account when designing the details of the interaction (Beaudouin-Lafon, 2004).

A critical role for an interactive system prototype is to illustrate how the user will interact with the system. While this may seem more natural with online prototypes, in fact it is often easier to explore different interaction strategies with offline prototypes. Note that interactivity and precision are orthogonal dimensions. One can create an imprecise prototype that is highly interactive, such as a series of paper screen images in which one person acts as the user and the other plays the system. One may create a very precise but noninteractive prototype, such as a detailed animation that shows feedback from a specific action by a user.

Prototypes can support interaction in various ways. For offline prototypes, one person (often with help from others) plays the role of the interactive system, presenting information and responding to the actions of another person playing the role of the user. For online prototypes, parts of the software are implemented, while others are played by a person. (This approach, called the "Wizard of Oz" after the character in the 1939 movie of the same name, is explained in a later section.) The key is that the prototype feels interactive to the user.

Prototypes can support different levels of interaction. Fixed prototypes, such as video clips or precomputed animations, are noninteractive: the user cannot interact, or pretend to interact, with it. Fixed prototypes are often used to illustrate or test scenarios. Fixed-path prototypes support limited interaction. The extreme case is a fixed prototype in which each step is triggered by a prespecified user action. For example, the person controlling the prototype might present the user with a screen containing a menu. When the user points to the desired item, he or she presents the corresponding screen showing a dialogue box. When the user points to the word "OK," he or she presents the screen that shows the effect of the command. Even though the position of the click is irrelevant (it is used as a trigger), the person playing the role of the user gets the feel of the interaction. Of course, this type of prototype can be much more sophisticated, with multiple options at each step. Fixed-path prototypes are very effective with scenarios and can be used for horizontal and task-based prototypes (see section on prototyping strategies below).

Open prototypes support large sets of interactions. Such prototypes work like the real system, with some limitations. They usually cover only part of the system (see vertical prototypes) and often have limited error handling or reduced performance relative to that of the final system.

Prototypes may thus illustrate or test different levels of interactivity. Fixed prototypes simply illustrate what the interaction might look like. Fixed-path prototypes provide designers and users with the experience of what the interaction might feel like, but only in prespecified situations. Open prototypes allow designers and users to explore a wide range of possible forms of interaction with the system.

Evolution

Prototypes have different life spans: rapid prototypes are created for a specific purpose and then thrown away, iterative prototypes evolve, either to work out some details (increasing their precision) or to explore various alternatives, and evolutionary prototypes are designed to become part of the final system.

Rapid prototypes are especially important in the early stages of design. They must be inexpensive and easy to produce, since the goal is to quickly explore a wide variety of possible types of interaction and then throw them away. Note that rapid prototypes may be offline or online. Creating precise software prototypes, even if they must be reimplemented in the final version of the system, is important for detecting and fixing interaction problems. A later section presents specific prototyping techniques, both offline and online.

Iterative prototypes are developed as a reflection of a design in progress, with the explicit goal of evolving through several design iterations. Designing prototypes that support evolution is sometimes difficult. There is a tension between evolving toward the final solution and exploring an unexpected design direction, which may be adopted or thrown away completely. Each iteration should inform some aspect of the design. Some iterations explore different variations of the same theme. Others may systematically increase precision, working out the finer details of the interaction. A later section describes tools and techniques for creating iterative prototypes.

Evolutionary prototypes are a special case of iterative prototypes in which the prototype evolves into part or all of the final system (Fig. 52.1). Obviously, this only applies to software prototypes. Extreme Programming (Beck, 2000) advocates this approach, tightly coupling design and implementation and building the system through constant evolution of its components. Evolutionary prototypes require more planning and practice than the approaches above because the prototypes are both representations of the final system and the final system itself, making it more difficult to explore alternative designs. We advocate a combined approach, beginning with rapid prototypes and then using iterative or evolutionary prototypes according to the needs of the project. A later section describes how to create iterative and evolutionary prototypes, using appropriate development tools.

PROTOTYPES AND THE DESIGN PROCESS

In the previous section, we looked at prototypes as artifacts, i.e., the results of a design process. Prototypes can also be seen as artifacts for design, as an integral part of the design process. Prototyping helps designers think: prototypes are the tools they use to solve design problems. In this section, we focus on prototyping as a process and its relationship to the overall design process.

User-Centered Design

The field of HCI is both user centered (Norman & Draper, 1986) and iterative. User-centered design places the user at the center of the design process, from the initial analysis of user requirements to testing and evaluation. Prototypes support this goal by allowing users see and experience the final system long before it is built. Designers can identify functional requirements,

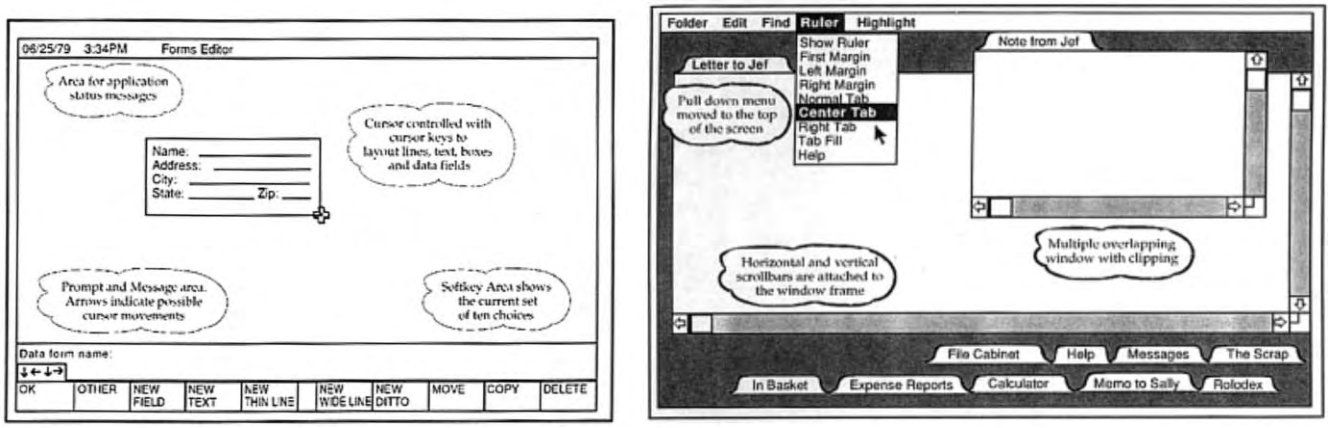


FIGURE 52.1. Evolutionary prototypes of the Apple Lisa: July 1979 (left), October 1980 (right) (Perkins et al., 1997) (© ACM, with permission).

usability problems and performance issues early and improve the design accordingly.

Iterative design involves multiple design-implement-test loops,[2] enabling the designer to generate different ideas and successively improve upon them. Prototypes support this goal by allowing designers to evaluate concrete representations of design ideas and select the best.

Prototypes reveal the strengths as well as the weaknesses of a design. Unlike pure ideas, abstract models or other representations, they can be contextualized to help understand how the real system would be used in a real setting. Because prototypes are concrete and detailed, designers can explore different real-world scenarios and users can evaluate them with respect to their current needs. Prototypes can be compared directly with other, existing systems, and designers can learn about the context of use and the work practices of the end users. Prototypes can help designers reanalyze the user's needs during the design process: not abstractly, as with traditional requirements analysis, but in the context of the system being built.

Participatory Design

Participatory design is a form of user-centered design that actively involves the user in all phases the design process (see Greenbaum & Kyng, 1991) Users are not simply consulted at the beginning and called in to evaluate the system at the end; they are treated as partners throughout. This early and active involvement of users helps designers avoid unpromising design paths and develop a deeper understanding of the actual design problem. Obtaining user feedback at each phase of the process also changes the nature of the final evaluation, which is used to fine tune the interface rather than discover major usability problems.

A common misconception about participatory design is that designers are expected to abdicate their responsibilities as designers, leaving the design to the end user. In fact, the goal is for designers and users to work together, each contributing their strengths to clarify the design problem as well as explore design solutions. Designers must understand what users can and cannot contribute. Usually, users are best at understanding the context in which the system will be used and subtle aspects of the problems that must be solved. Innovative ideas can come from both users and designers, but the designer is responsible for considering a wide range of options that might not be known to the user and balancing the trade-offs among them.

Because prototypes are shared, concrete artifacts, they serve as an effective medium for communication within the design team as well as with users. We have found that collaborating on prototype design is an effective way to involve users in participatory design. Prototypes help users articulate their needs and reflect on the efficacy of design solutions proposed by designers.

Exploring the Design Space

Design is not a natural science: the goal is not to describe and understand existing phenomena but to create something new. Of course, designers benefit from scientific research findings, and they may use scientific methods to evaluate interactive systems. However, designers also require specific techniques for generating new ideas and balancing complex sets of trade-offs to help them develop and refine design ideas.

Designers from fields such as architecture and graphic design have developed the concept of a design space, which constrains design possibilities along some dimensions, while leaving others open for creative exploration. Ideas for the design space come from many sources: existing systems, other designs, other designers, external inspiration, and accidents that prompt new ideas. Designers are responsible for creating a design space specific to a particular design problem. They explore this design space, expanding and contracting it as they add and eliminate ideas. The

[2]Software engineers refer to this as the Spiral model (Boehm, 1988).

process is iterative: more cyclic than reductionist. That is, the designer does not begin with a rough idea and successively add more details that are precise until the final solution is reached. Instead, he or she begins with a design problem, which imposes a set of constraints, and then generates a set of ideas to form the initial design space. He or she then explores this design space, preferably with the user, and selects a particular design direction to pursue. This closes off part of the design space, but opens up new dimensions that can be explored. The designer generates additional ideas along these dimensions, explores the expanded design space, and then makes new design choices. Design principles (e.g., Beaudouin-Lafon & Mackay, 2000) help this process by guiding it both in the exploration and choice phases. The process continues, in a cyclic expansion and contraction of the design space, until a satisfying solution is reached or time has run out.

All designers work with constraints: not just limited budgets and programming resources, but also design constraints. These are not necessarily bad: one cannot be creative along all dimensions at once. However, some constraints are unnecessary, derived from poor framing of the original design problem. If we consider a design space as a set of ideas and a set of constraints, the designer has two options. She can modify ideas within the specified constraints or modify the constraints to enable new sets of ideas. Unlike traditional engineering, which treats the design problem as a given, designers are encouraged to challenge, and if necessary, change the initial design problem. If she reaches an impasse, the designer can either generate new ideas or redefine the problem (and thus change the constraints). Some of the most effective design solutions derive from a more careful understanding and reframing of the design brief.

Note that all members of the design team, including users, may contribute ideas to the design space and help select design directions from within it. However, it is essential that these two activities are kept separate. Expanding the design space requires creativity and openness to new ideas. During this phase, everyone should avoid criticizing ideas and concentrate on generating as many as possible. Clever ideas, half-finished ideas, silly ideas, and impractical ideas all contribute to the richness of the design space and improve the quality of the final solution. In contrast, contracting the design space requires critical evaluation of ideas. During this phase, everyone should consider the constraints and weigh the trade-offs. Each major design decision must eliminate part of the design space: rejecting ideas is necessary in order to experiment and refine others and make progress in the design process. Choosing a particular design direction should spark new sets of ideas, and those new ideas are likely to pose new design problems. In summary, exploring a design space is the process of moving back and forth between creativity and choice.

Prototypes aid designers in both aspects of working with a design space: generating concrete representations of new ideas and clarifying specific design directions. The next two sections describe techniques that have proven most useful in our own prototyping work, both for research and product development.

Expanding the Design Space: Generating Ideas

The most well-known idea generation technique is brainstorming, introduced by Osborn (1957). His goal was to create synergy within the members of a group: ideas suggested by one participant would spark ideas in other participants. Subsequent studies (Collaros & Anderson, 1969; Diehl & Stroebe, 1987) challenged the effectiveness of group brainstorming, finding that aggregates of individuals could produce the same number of ideas as groups. They found certain effects, such as production blocking, free riding, and evaluation apprehension, were sufficient to outweigh the benefits of synergy in brainstorming groups. Since then, many researchers have explored different strategies for addressing these limitations. For our purpose, the quantity of ideas is not the only important measure: the relationships among members of the group are also important. As de Vreede et al. (2000) pointed out, one should also consider elaboration of ideas, as group members react to each other's ideas.

We have found that brainstorming, including a number of variants, is an important group-building exercise for participatory design. Of course, designers may brainstorm ideas by themselves. However, brainstorming in a group is more enjoyable and, if it is a recurring part of the design process, plays an important role in helping group members share and develop ideas together.

The simplest form of brainstorming involves a small group of people. The goal is to generate as many ideas as possible on a prespecified topic: quantity, not quality, is important. Brainstorming sessions have two phases: the first for generating ideas and the second for reflecting upon them. The initial phase should last no more than an hour. One person should moderate the session, keeping time, ensuring that everyone participates, and preventing people from critiquing each other's ideas. Discussion should be limited to clarifying the meaning of a particular idea. A second person records every idea, usually on a flipchart or transparency on an overhead projector. After a short break, participants are asked to reread all the ideas and each person marks their three favorite ideas.

One variation is designed to ensure that everyone contributes, not just those who are verbally dominant. Participants write their ideas on individual cards or Post-it notes for a prespecified period. The moderator then reads each idea aloud. Authors are encouraged to elaborate (but not justify) their ideas, which are then posted on a whiteboard or flipchart. Group members may continue to generate new ideas, inspired by the others they hear.

We use a variant of brainstorming, called "video brainstorming" (Mackay, 2000), as a very fast technique for prototyping interaction: instead of simply writing or drawing their ideas, participants act them out in front of a video camera (Fig. 52.2). The goal is the same as other brainstorming exercises, i.e., to create as many new ideas as possible, without critiquing them. However, the use of video, combined with paper or cardboard mock ups, encourages participants to experience the details of the interaction and to understand each idea from the perspective of the user, while preserving a tangible record of the idea.

Each video brainstorming idea should take two to five minutes to generate and capture, allowing participants to simulate a wide variety of ideas very quickly. The resulting video clips provide illustrations of each idea that are easier to understand (and remember) than hand-written notes. (We find that raw notes from brainstorming sessions are not very useful after a few weeks because the participants no longer remember the con-

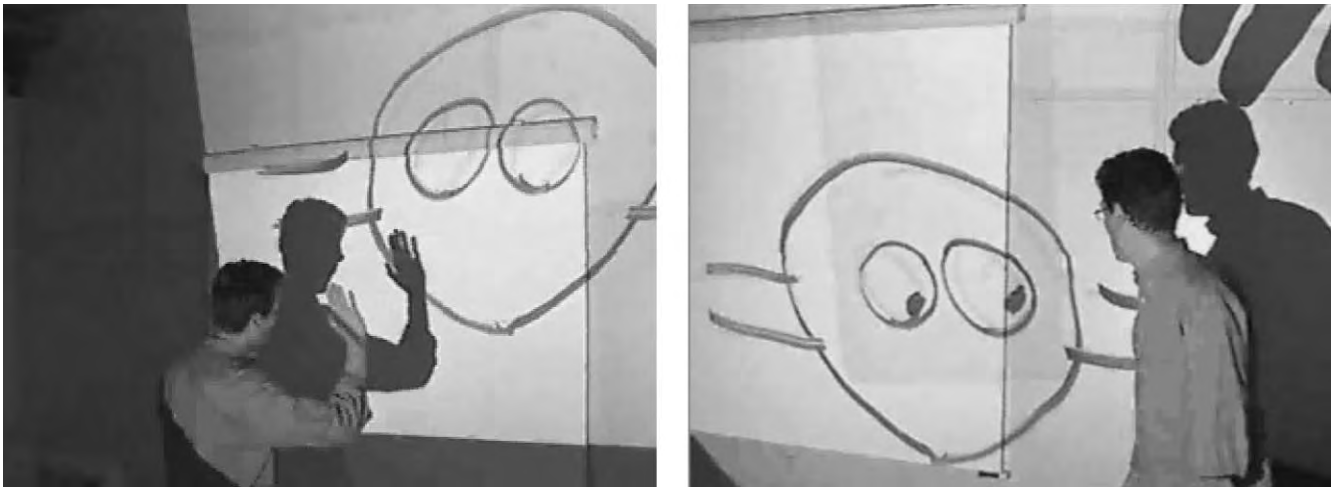


FIGURE 52.2. Video Brainstorming an animated character: One participant uses an overhead projector to project an image on the wall and responds to the actions of a second participant, who plays the role of the user. Here, the animated character, a very rough sketch on a transparency, reponds when the user waves and moves its eyes to follow the user.

text in which the ideas were created, whereas videobrainstorming clips are useful years later.)

Video brainstorming requires thinking more deeply about each idea than in traditional oral brainstorming. It is possible to stay vague and general when describing an interaction in words or even with a sketch, but acting out the interaction in front of the camera forces the author of the idea (and the other participants) to consider seriously the details of how a real user would actually interact with the idea. Video brainstorming also encourages designers and users to think about new ideas in the context in which they will be used. We also find that video clips from a video brainstorming session, even though rough, are much easier for the design team to interpret than written ideas from a standard brainstorming session.

We generally run a standard brainstorming session, either orally or with cards, prior to a video brainstorming session, in order to maximize the number of ideas to be explored. Participants then take their favorite ideas from the previous session and develop them further as video brainstorms. Each person is asked to direct at least two ideas, incorporating the hands or voices of other members of the group. We find that, unlike standard brainstorming, video brainstorming encourages even the quietest team members to participate.

Contracting the Design Space: Selecting Alternatives

After expanding the design space by creating new ideas, designers must stop and reflect upon the choices available to them. After exploring the design space, designers must evaluate their options and make concrete design decisions: choosing some ideas, specifically rejecting others, and leaving other aspects of the design open to further idea generation activities. Rejecting good, potentially effective ideas is difficult, but necessary to make progress.

Prototypes often make it easier to evaluate design ideas from the user's perspective. They provide concrete representations that can be compared. Many of the evaluation techniques described elsewhere in this handbook could be applied to prototypes, to help focus the design space. The simplest situation is when the designer must choose among several discrete, independent options. Running a simple experiment, using techniques borrowed from psychology, allows the designer to compare how users respond to each of the alternatives. The designer builds a prototype, with either fully implemented or simulated versions of each option. The next step is to construct tasks or activities that are typical of how the system would be used, and ask people from the user population to try each of the options under controlled conditions. It is important to keep everything the same, except for the options being tested.

Designers should base their evaluations on both quantitative measures, such as speed or error rate, and qualitative measures, such as the user's subjective impressions of each option. Ideally, of course, one design alternative will be clearly faster, prone to fewer errors, and preferred by the majority of users. More often, the results are ambiguous, and the designer must consider other factors when making the design choice. (Interestingly, running small experiments often highlights other design problems and may help the designer reformulate the design problem or change the design space.)

The more difficult (and common) situation, is when the designer faces a complex, interacting set of design alternatives, in which each design decision affects a number of others. Designers can use heuristic evaluation techniques, which rely on our understanding of human cognition, memory, and sensory-perception. They can also evaluate their designs with respect to ergonomic criteria or design principles (Beaudouin-Lafon & Mackay, 2000).

Another strategy is to create one or more scenarios that illustrate how the combined set of features will be used in a realistic

setting. The scenario must identify who is involved, where the activities take place, and what the user does over a specified period. Good scenarios involve more than a string of independent tasks; they should incorporate real-world activities, including common or repeated tasks, successful activities, breakdowns, and errors, with both typical and unusual events. The designer then creates a prototype that simulates or implements the aspects of the system necessary to illustrate each set of design alternatives. Such prototypes can be tested by asking users to walk through the same scenario several times, once for each design alternative. As with experiments and usability studies, designers can record both quantitative and qualitative data, depending on the level of the prototypes being tested.

The previous section described an idea-generation technique called "video brainstorming," which allows designers to generate a variety of ideas about how to interact with the future system. We call the corresponding technique for focusing in on a design "video prototyping." Video prototyping can incorporate any of the rapid-prototyping techniques (offline or online) described in a later section. Video prototypes are quick to build, force designers to consider the details of how users will react to the design in the context in which it will be used, and provide an inexpensive method of comparing complex sets of design decisions.

To an outside observer, video brainstorming and video prototyping techniques look very similar: both involve small design groups working together, creating rapid prototypes, and interacting with them in front of a video camera. Both result in video illustrations that make abstract ideas concrete and help team members communicate with each other. The critical difference is that video brainstorming expands the design space, by creating a number of unconnected collections of individual ideas, whereas video prototyping contracts the design space, by showing how a specific collection of design choices work together in a single design proposal.

Prototyping Strategies

Designers must decide what role prototypes should play with respect to the final system and in which order to create different aspects of the prototype. The next section presents four strategies: horizontal, vertical, task oriented, and scenario based, which focus on different design concerns. These strategies can use any of the prototyping techniques covered in the following sections.

Horizontal prototypes. The purpose of a horizontal prototype is to develop one entire layer of the design at the same time. This type of prototyping is most common with large software development teams, where designers with different skill sets address different layers of the software architecture. Horizontal prototypes of the user interface are useful to get an overall picture of the system from the user's perspective and address issues such as consistency (similar functions are accessible through similar user commands), coverage (all required functions are supported), and redundancy (the same function is/is not accessible through different user commands).

User interface horizontal prototypes can begin with rapid prototypes and progress through to working code. Software prototypes can be built with an interface builder without creating any of the underlying functionality, making it possible to test how the user will interact with the user interface without worrying about how the rest of the architecture works. However, some level of scaffolding or simulation of the rest of the application is often necessary; otherwise, the prototype cannot be evaluated properly. Consequently, software horizontal prototypes tend to be evolutionary, i.e., they are progressively transformed into the final system.

Vertical prototypes. The purpose of a vertical prototype is to ensure that the designer can implement the full, working system, from the user interface layer down to the underlying system layer. Vertical prototypes are often built to assess the feasibility of a feature described in a horizontal, task-oriented, or scenario-based prototype. For example, when we developed the notion of magnetic guidelines in the CPN2000 system to facilitate the alignment of graphical objects (Beaudouin-Lafon & Mackay, 2000, Beaudouin-Lafon & Lassen, 2000), we implemented a vertical prototype to test not only the interaction technique but also the layout algorithm and the performance. We knew that we could only include the particular interaction technique if we could implement a sufficiently fast response.

Vertical prototypes are generally high precision software prototypes because their goals are to validate ideas at the system level. They are often thrown away because they are generally created early in the project, before the overall architecture has been decided, and they focus on only one design question. For example, a vertical prototype of a spelling checker for a text editor does not require text-editing functions to be implemented and tested. However, the final version will need to be integrated into the rest of the system, which may involve considerable architectural or interface changes.

Task-oriented prototypes. Many user interface designers begin with a task analysis to identify the individual tasks that the user must accomplish with the system. Each task requires a corresponding set of functionality from the system. Task-based prototypes are organized as a series of tasks, which allows both designers and users to test each task independently, systematically working through the entire system.

Task-oriented prototypes include only the functions necessary to implement the specified set of tasks. They combine the breadth of horizontal prototypes, to cover the functions required by those tasks, with the depth of vertical prototypes, enabling detailed analysis of how the tasks can be supported. Depending on the goal of the prototype, both offline and online representations can be used for task-oriented prototypes.

Scenario-based prototypes. Scenario-based prototypes are similar to task-oriented ones, except that they do not stress individual, independent tasks, but rather follow a more realistic scenario of how the system would be used in a real-world setting. Scenarios are stories that describe a sequence of events and how the user reacts. A good scenario includes both common and unusual situations, and should explore patterns of activity over time. Bødker (1995) developed a checklist to ensure that no important issues have been left out.

We find it useful to begin with anecdotes derived from observations of or interviews with real users. Ideally, some of those users should participate in the creation of specific use and other users should critique them based on how realistic they are. Use scenarios are then turned into design scenarios, in which the same situations are described but with the functionality of the proposed system. Design scenarios are used, among other things, to create scenario-based video prototypes or software prototypes. Like task-based prototypes, the developer needs to write only the software necessary to illustrate the components of the design scenario. The goal is to create a situation in which the user can experience what the system would be like in a realistic use context, even if it addresses only a subset of the planned functionality.

The following section describes a variety of rapid prototyping techniques that can be used in any of these four prototyping strategies. We begin with offline rapid prototyping techniques, followed by online prototyping techniques.

RAPID PROTOTYPING

The goal of rapid prototyping is to develop prototypes very quickly, in a fraction of the time it would take to develop a working system. By shortening the prototype-evaluation cycle, the design team can evaluate more alternatives and iterate the design several times, improving the likelihood of finding a solution that successfully meets the user's needs. Rapid prototypes also serve to cut off unpromising design directions, saving time, and money. It is far easier to reject an idea based on a rapid prototype than a more fully developed one, and one reduces the chance of spending a great deal of time and effort on a design that ultimately does not work.

How rapid is rapid depends on the context of the particular project and the stage in the design process. Early prototypes, such as sketches, can be created in a few minutes. Later in the design cycle, a prototype produced in less than a week may still be considered rapid if the final system is expected to take months or years to build. Precision, interactivity, and evolution all affect the time it takes to create a prototype. Not surprisingly, a precise and interactive prototype takes more time to build than an imprecise or fixed one.

The techniques presented in this section are organized from most rapid to least rapid, according to the representation dimension previously introduced. Offline techniques are generally more rapid than online techniques. However, creating successive iterations of an online prototype may end up being faster than creating new offline prototypes.

Offline Rapid Prototyping Techniques

Offline prototyping techniques range from simple to very elaborate. Because they do not involve software, they are usually considered a tool for thinking through the design issues, to be thrown away when they are no longer needed. This section describes simple paper and pencil sketches, three-dimensional mock ups, Wizard of Oz simulations, and video prototypes.

Paper and pencil. The fastest form of prototyping involves paper, transparencies and Post-it notes to represent aspects of an interactive system (e.g., Muller, 1991). By playing the roles of both the user and the system, designers can get a quick idea of a wide variety of different layout and interaction alternatives, in a very short period of time (Rettig, 1994; Snyder, 2003).

Designers can create a variety of low-cost special effects. For example, a tiny triangle drawn at the end of a long strip cut from an overhead transparency makes a handy mouse pointer, which can be moved by a colleague in response to the user's actions. Post-it notes, with prepared lists, can provide pop-up menus. An overhead projector pointed at a whiteboard makes it easy to project transparencies (hand drawn or preprinted, overlaid on each other as necessary) to create an interactive display on the wall. The user can interact by pointing (Fig. 52.3) or drawing on the whiteboard. One or more people can watch the user and move the transparencies in response to her actions. Everyone in the room gets an immediate impression of how the eventual interface might look and feel.

Note that most paper prototypes begin with quick sketches on paper, then progress to more carefully drawn screen images made with a computer (Fig. 52.4). In the early stages, the goal is to generate a wide range of ideas and expand the design space, not to determine the final solution. Paper and pencil prototypes are an excellent starting point for horizontal, task-based, and scenario-based prototyping strategies.

Mock ups. Architects use mock ups or scaled prototypes to provide three-dimensional illustrations of future buildings. Mock ups are also useful for interactive system designers, helping them move beyond two-dimensional images drawn on paper or transparencies (see Bødker, Ehn, Knudsen, Kyng, & Madsen, 1988). Generally made of cardboard, foamcore, or other found materials (Frishberg, 2006), mock ups are physical prototypes of



FIGURE 52.3. Hand-drawn transparencies can be projected onto a wall, creating an interface a user can respond to.

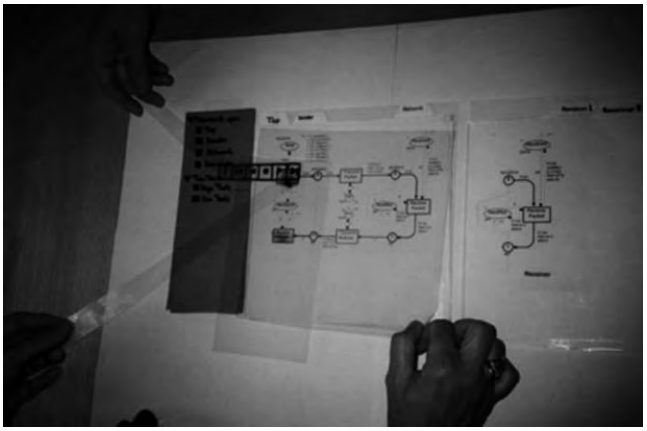


FIGURE 52.4. Several people work together to simulate interacting with this paper prototype. One person moves a transparency with a mouse pointer while another moves the diagram accordingly.

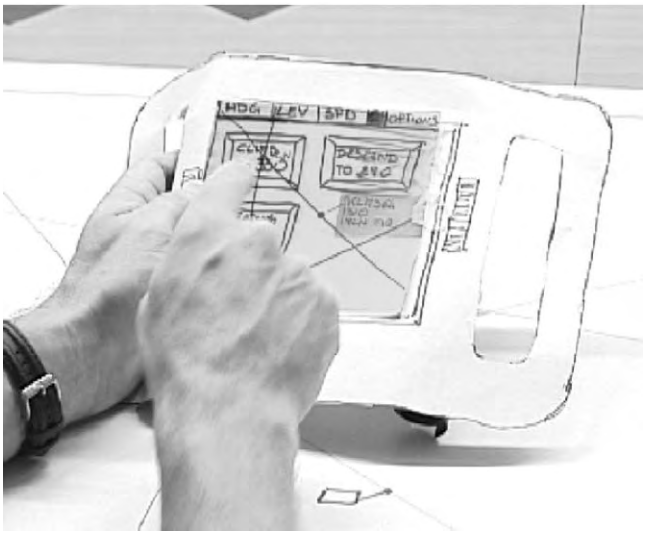


FIGURE 52.5 Mock-up of a hand-held display with carrying handle.

the new system. Fig. 52.5 shows an example of a handheld mock up showing the interface to a new handheld device. The mock up provides a deeper understanding of how the interaction will work in real-world situations than a set of screen images.

Mock ups allow the designer to concentrate on the physical design of the device, such as the position of buttons or the screen. The designer can also create several mock ups and compare input or output options, such as buttons versus trackballs. Designers and users should run through different scenarios, identifying potential problems with the interface or generating ideas for new functionality. Mock ups can also help the designer envision how an interactive system will be incorporated into a physical space (Fig. 52.6).

Wizard of Oz. Sometimes it is useful to give users the impression that they are working with a real system, even before it exists. Kelley (1983) dubbed this technique the *Wizard of Oz*,

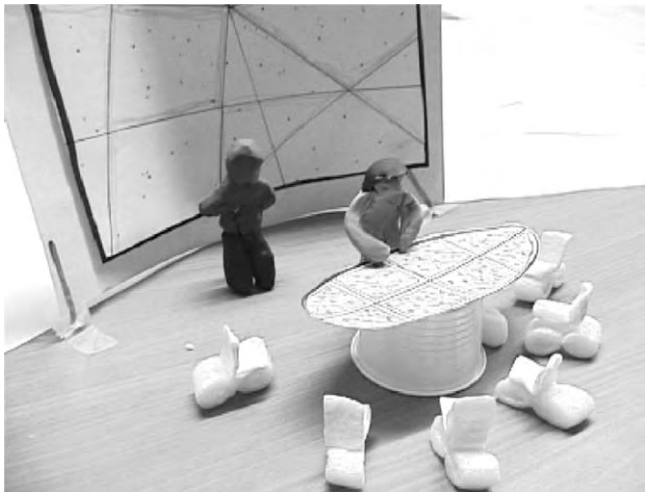


FIGURE 52.6. Scaled mock-up of an air traffic control table, connected to a wall display.

based on the scene in the 1939 movie of the same name. The heroine, Dorothy, and her companions ask the mysterious Wizard of Oz for help. When they enter the room, they see an enormous green human head breathing smoke and speaking with a deep, impressive voice. When they return later, they again see the Wizard. This time, Dorothy's small dog pulls back a curtain, revealing a frail old man pulling levers and making the mechanical Wizard of Oz speak. They realize that the impressive being before them is not a wizard at all, but simply an interactive illusion created by the old man.

The software version of the Wizard of Oz operates on the same principle. A user sits a screen and interacts with what appears to be a working program. Hidden elsewhere, the software designer (the Wizard) watches what the user does and, by responding to different user actions, creates the illusion of a working software program. In some cases, the user is unaware that a person, rather than a computer, is operating the system.

The Wizard of Oz technique lets users interact with partially functional computer systems. Whenever they encounter something that has not been implemented (or there is a bug), a human developer who is watching the interaction overrides the prototype system and plays the role destined to eventually be played by the computer. A combination of video and software can work well, depending upon what needs to be simulated.

The Wizard of Oz technique was initially used to develop natural language interfaces (e.g., Chapanis, 1982; Good et al., 1984). Since then, the technique has been used in a wide variety of situations, particularly those in which rapid responses from users are not critical. Wizard of Oz simulations may consist of paper prototypes, fully implemented systems, and everything in between.

Video prototyping. Video prototypes (Mackay, 1988) use video to illustrate how users will interact with the new system. As explained in an earlier section, they differ from video brainstorming in that the goal is to refine a single design, not generate new ideas. Video prototypes may build upon paper and pen-

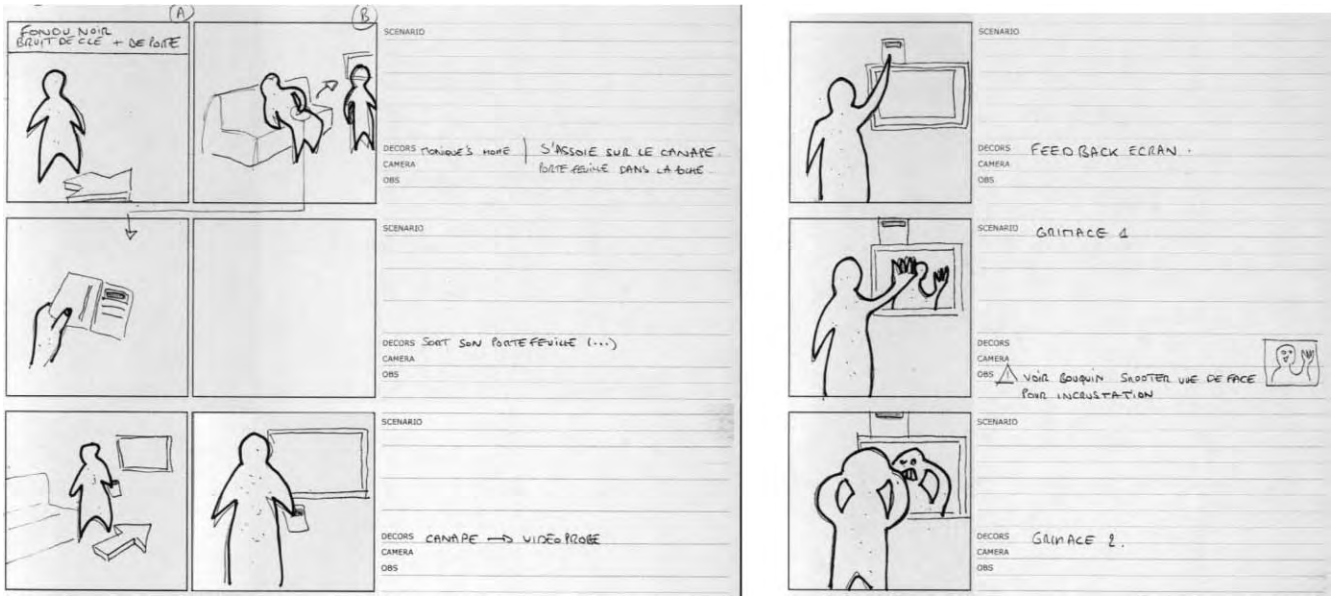


FIGURE 52.7. A storyboard for a tangible interface that enables users to establish and manage their connections to a small group of friends and family.

cil prototypes and cardboard mock ups and can use existing software and images of real-world settings.

We begin our video prototyping exercises by reviewing relevant data about users and their work practices, and then review ideas we video brainstormed. The next step is to create a use scenario, describing the user at work. Once the scenario is described in words, the designer develops a storyboard. Similar to a comic book, the storyboard shows a sequence of rough sketches of each action or event, with accompanying actions and/or dialogue (or subtitles), with related annotations that explain what is happening in the scene or the type of shot (Fig. 52.7). A paragraph of text in a scenario corresponds to about a page of a storyboard.

Storyboards help designers refine their ideas, generate "what- if" scenarios for different approaches to a story, and communicate with the other people who are involved in the design process. Some storyboards may be informal sketches of ideas, with only partial information. Others follow a predefined format and are used to direct the production and editing of a video prototype. Designers should jot down notes on storyboards as they think through the details of the interaction.

Paper storyboards can be used as is to communicate with other members of the design team. Designers and users can discuss the proposed system and alternative ideas for interacting with it (Fig. 52.8). Simple videos of each successive frame, with a voice over to explain what happens, can also be effective. We usually use storyboards to help us shoot video prototypes, which illustrate how a new system will look to a user in a real-world setting. We find that placing the elements of a storyboard on separate cards and arranging them (Mackay, & Pagani, 1994) helps the designer experiment with different linear sequences and insert or delete video clips. Continuing to the next step, i.e. creating a video prototype based on the storyboard, forces designers to consider the design details in even greater depth.



FIGURE 52.8 Video Prototyping: The CPN design team reviews their observations of CPN developers and then discuss several design alternatives. They work out a scenario and storyboard it, then shoot a video prototype that reflects their design.

Storyboards, even very informal ones, are essential guides for shooting video prototypes. To avoid spending time in post-production, we use a technique called "editing-in-the-camera" (see Mackay, 2000) in which each video clip, guided by the storyboard, is shot in the order that it will be viewed. With a well-designed storyboard, this is just as easy and results in a finished video prototype at the end of a one-hour session. Note that today's digital video cameras include editing features in the camera, which introduces the temptation to make editing changes on the fly. Our students who do this consistently take more time than their colleagues who do not, usually with worse results. In general, we recommend avoiding post hoc editing and just following the storyboard.

We use title cards, as in a silent movie, to separate the clips. This both simplifies shooting and makes it easier for the viewer to follow the story. Title cards also provide the only acceptable way to edit while you are shooting: if you make an error, you should address it immediately by rewinding to the last title card and continue shooting from there.

Video prototypes take several forms. In some, a narrator explains each event and several people on the sidelines may be necessary to move images and illustrate the interaction. In others, actors simply perform the movements and the viewer is expected to understand the interaction without a voice over. You can easily create simple special effects. For example, time-lapse photography allows you to have images appear and disappear based on the user's interaction. For example, record a clip of a user pressing a button, press "pause" on the camera, add a pop-up menu, then restart the camera, to create the illusion of immediate system feedback.

Video prototypes should begin with a title, followed by an establishing shot that shows the user in the context defined by the scenario. Next, create a series of close-up and midrange shots, interspersed with title cards to explain the story. Place a final title card with credits at the end. We print blank title cards on colored paper to make it easier to search for the sections of the video later: When you fast-forward through video, a solid blue or red frame clearly stands out (for detailed examples of video-based prototyping techniques, see Mackay, 2002).

Video prototypes are fixed prototypes. However, it is also possible to use video as an open prototype, in which users can interact with the prototype in an open-ended way. Video thus becomes a tool for sketching and visualizing interactions. For example, we sometimes use a live video camera as a Wizard of Oz tool and capture the interaction with a second video camera. The Wizard should have access to a set of prototyping materials representing screen objects. Other team members stand by ready to help move objects as needed. The live camera is pointed at the Wizard's work area, with either a paper prototype or a partially working software simulation. The resulting image is projected onto a screen or monitor in front of the user. One or more people should be situated so that they can observe the actions of the user and manipulate the projected video image accordingly. This is most effective if the Wizard is well prepared for a variety of events and can present semiautomated information. The user interacts with the objects he or she sees on the screen and the Wizard moves the relevant materials in direct response to each user action. The other camera records the interaction between the user and the simulated software system on the screen, to create either a video brainstorm (for a quick idea) or a fully storyboarded video prototype.

Fig. 52.9 shows a Wizard of Oz simulation with a live video camera, video projector, whiteboard, overhead projector, and transparencies. The setup allows two people to experience how they would communicate via a new interactive communication system. One video camera films the blond woman, who can see and talk to the brunette woman. Her image is projected live onto the left side of the wall. An overhead projector displays hand drawn transparencies, manipulated by two other people, in response to gestures made by the brunette woman. The entire interaction is videotaped by a second video camera. Note that participants at a workshop on user interfaces for air-traffic control created this video: none of the participants had ever used video prototyping techniques but they were able to set up this Wizard of Oz style environment and use it to generate new interaction ideas in less than 30 minutes.



FIGURE 52.9. Complex wizard-of-Oz simulation, with projected image from a live video camera and transparencies projected from an overhead projector.

Combining Wizard of Oz and video is a particularly powerful prototyping technique because it gives the person playing the user a real sense of what it might actually feel like to interact with the proposed tool, long before it has been implemented. Seeing a video clip of someone else interacting with a simulated tool is more effective than simply hearing about it, but interacting with it directly is more powerful still. Note that video should be used with caution, particularly when video prototypes are taken out of their original design setting (for a more detailed discussion of the ethical issues involved, see Mackay, 1995).

Video prototyping may act as a form of specification for developers, enabling them to build the precise interface, both visually and interactively, created by the design team. This is particularly useful when moving from offline to online prototypes, which we now describe.

Online Rapid Prototyping Techniques

The goal of online rapid prototyping is to create higher precision prototypes than can be achieved with offline techniques. Such prototypes may prove useful to better communicate ideas to clients, managers, developers and end users. They are also useful for the design team to fine tune the details of a layout or an interaction. They may exhibit problems in the design that were not apparent in less precise prototypes. Finally, they may be used early on in the design process for low precision prototypes that would be difficult to create offline, such as when very dynamic interactions or visualizations are needed.

The techniques presented in this section are sorted by interactivity. We start with noninteractive simulations, such as animations, followed by interactive simulations that provide fixed or multiple-paths interactions. We finish with scripting languages, which support open interactions.

Noninteractive simulations. A noninteractive simulation is a computer-generated animation that represents what a person would see of the system if he or she were watching over the user's shoulder. Noninteractive simulations are usually created when offline prototypes, including video, fail to capture a particular aspect of the interaction and it is important to have a quick prototype to evaluate the idea. It is usually best to start by creating a storyboard to describe the animation, especially if the developer of the prototype is not a member of the design team.

One of the most widely used tools for noninteractive simulations is Macromedia Director™. The designer defines graphical objects called "sprites," and defines paths along which to animate them. The succession of events, such as when sprites appear and disappear, is determined with a time line. Sprites are usually created with drawing tools, such as Adobe Illustrator or Deneba Canvas, painting tools, such as Adobe Photoshop, or even scanned images. Director is a very powerful tool and experienced developers use it quickly to create sophisticated interactive simulations. (However, it is still faster to create noninteractive simulations.) Other similar tools exist on the market such as Adobe AfterEffects and Macromedia Flash (Fig. 52.10).

Fig. 52.11 shows a set of animation movies created by Curbow to explore the notion of accountability in computer sys-

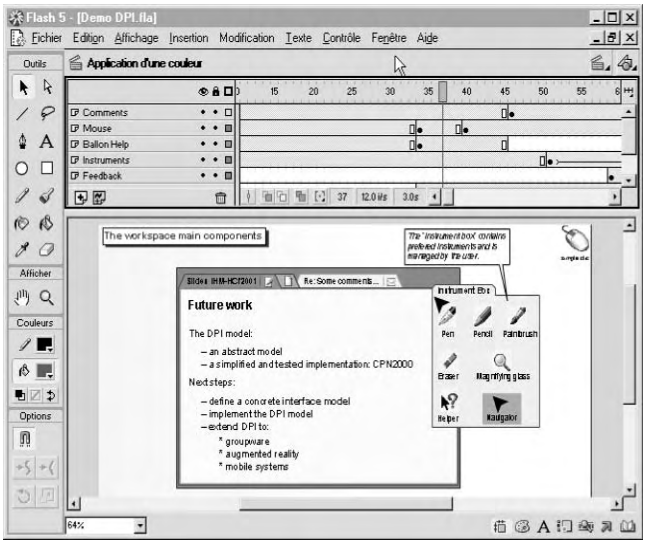


FIGURE 52.10. A non-interactive simulation of a desktop interface created with Macromedia Flash. The time-line (top) displays the active sprites while the main window (bottom) shows the animation. (© O. Beaudoux, with permission)

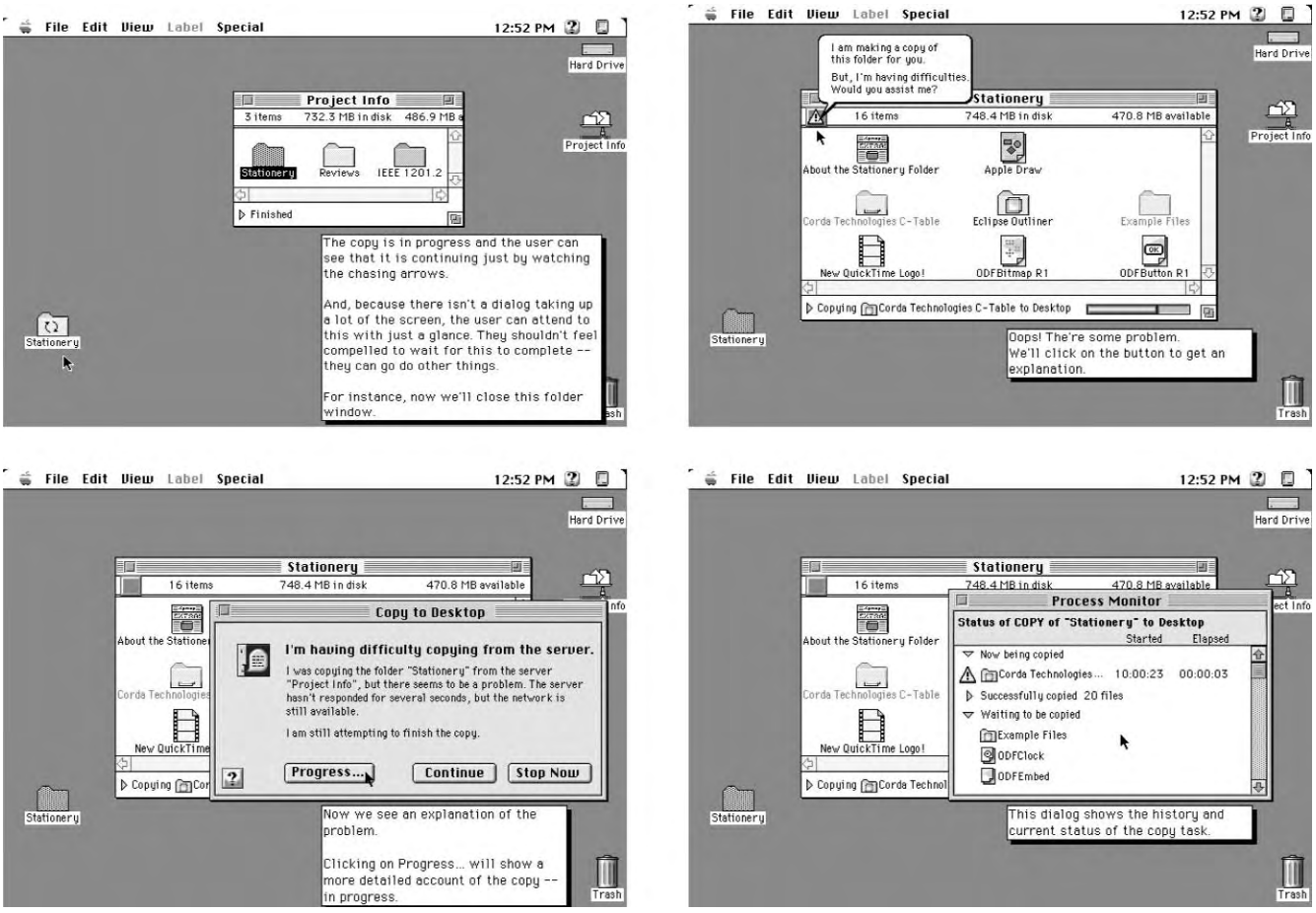


FIGURE 52.11. Frames from an animated simulation created with Macromind Director (© D. Curbow, with permission)

tems (Dourish, 1997). These prototypes explore new ways to inform the user of the progress of a file copy operation. They were created with Macromind Director by combining custom-made sprites with sprites extracted from snapshots of the Macintosh Finder. The simulation features cursor motion, icons being dragged, windows opening and closing, and so forth. The result is a realistic prototype that shows how the interface looks and behaves, that was created in just a few hours. Note that the simulation also features text annotations to explain each step, which helps document the prototype.

Noninteractive animations can be created with any tool that generates images. For example, many web designers use Adobe Photoshop to create simulations of their websites. Photoshop images are composed of various layers that overlap like transparencies. The visibility and relative position of each layer can be controlled independently. Designers can quickly add or delete visual elements, simply by changing the characteristics of the relevant layer. This permits quick comparisons of alternative designs and helps visualize multiple pages that share a common layout or banner. Skilled Photoshop users find this approach much faster than most web-authoring tools.

We used this technique in the CPN2000 project (Mackay, Ratzer, & Janecek, 2000) to prototype the use of transparency. After several prototyping sessions with transparencies and overhead projectors, we moved to the computer to understand the differences between the physical transparencies and the transparent effect as it would be rendered on a computer screen. We later developed an interactive prototype with OpenGL, which required an order of magnitude more time to implement than the Photoshop mock up.

Even a spreadsheet program can be used for prototyping: Berger (2006) described the use of Microsoft Excel to prototype form-based interfaces. First, a template is created that contains a number of reusable elements by taking advantage of the workbook feature of Excel, where multiple pages can be presented with a tabbed interface. Then, prototypes are created by copying and pasting items from the template into a blank page, taking advantage of the table structure of the spreadsheet to create grid layouts.

Interactive simulations. Designers can also use tools such as Adobe Photoshop to create Wizard of Oz simulations. For example, the effect of dragging an icon with the mouse can be obtained by placing the icon of a file in one layer and the icon of the cursor in another layer and by moving either or both layers. The visibility of layers, as well as other attributes, can also create more complex effects. Like Wizard of Oz and other paper prototyping techniques, the behavior of the interface is generated by the user who is operating the Photoshop interface.

More specialized tools, such as Hypercard and Macromedia Director, can be used to create simulations with which the user can directly interact. Hypercard (Goodman, 1987) was one of the most successful early prototyping tools. It was an authoring environment based on a stack metaphor: a stack contains a set of cards that share a background, including fields and buttons. Each card can also have its own unique contents, including fields and buttons (Fig. 52.12). Using a scripting language, stacks, cards, fields, and buttons can react to user events, such as clicking a button, as well as system events, for example, when a new card is displayed or about to disappear.

Interfaces can be prototyped quickly with this approach, by drawing different states in successive cards and using buttons to switch from one card to the next. Multiple-path interactions can be programmed by using several buttons on each card. Interactions that are more open require advanced use of the scripting language, but are easy to master with a little practice.

Macromind Director uses a different metaphor, attaching behaviors to sprites and to frames of the animation. For example, a button can be defined by attaching a behavior to the sprite representing that button. When the sprite is clicked, the animation jumps to a different sequence. This is usually coupled with a behavior attached to the frame containing the button that loops the animation on the same frame. As a result, nothing happens until the user clicks the button, at which point the animation skips to a sequence where, for example, a dialogue box opens. The same technique can be used to make the OK and Cancel buttons of the dialogue box interactive. Typically, the Cancel button would skip to the original frame while the OK button would skip to a third sequence. Director comes with a large library of behaviors to describe such interactions, so that prototypes can be created completely interactively. New behaviors can also be defined with a scripting language called Lingo.

Scripting languages. Scripting languages are the most advanced rapid prototyping tools. As with the interactive-simulation tools described above, the distinction between rapid prototyping tools and development tools is not always clear. Scripting languages make it easy to quickly develop throwaway prototypes (a few hours to a few days), which may or may not be used in the final system, for performance or other technical reasons.

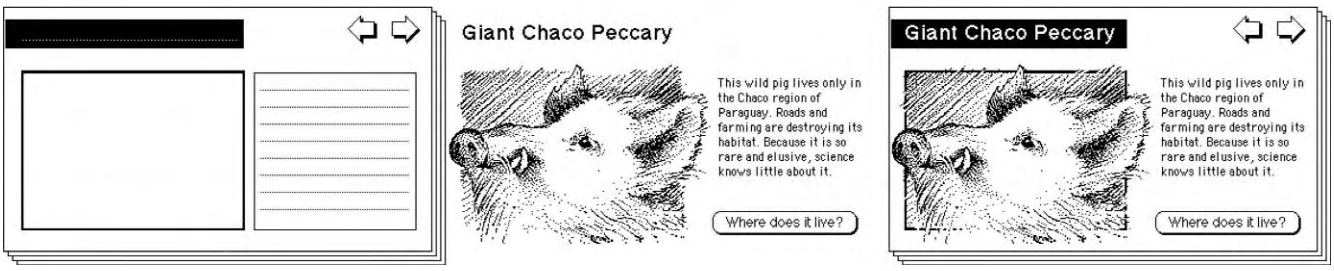


FIGURE 52.12. A Hypercard card (right) is the combination of a background (left) and the card's content (center) (© Apple Computer, with permission)

A scripting language is a programming language that is both lightweight and easy to learn. Most scripting languages are interpreted or semicompiled; for example, the user does not need to go through a compile-link-run cycle each time the script (program) is changed. Scripting languages can be forbidding: they are not strongly typed and nonfatal errors are ignored unless explicitly trapped by the programmer. Scripting languages are often used to write small applications for specific purposes and can serve as glue between preexisting applications or software components.

Tcl (Ousterhout, 1993) is particularly suitable to develop user interface prototypes (or small to medium-size applications) because of its Tk user interface toolkit. Tcl was inspired by the syntax of the Unix shell and makes it very easy to interface existing applications by turning the application programming interface (API) into a set of commands that can be called directly from a Tcl script. Tk features all the traditional interactive objects (called "widgets") of a UI toolkit: buttons, menus, scrollbars, lists, dialogue boxes, and so forth. A widget is typically created with only one line. For example,

```
button .dialogbox.ok -text OK -command {de-
stroy .dialogbox}.
```

This command creates a button, called ".dialogbox.ok," whose label is "OK." It deletes its parent window ".dialogbox" when the button pressed. A traditional programming language and toolkit would take 5 to 20 lines to create the same button.

Tcl also has two advanced, heavily parameterized widgets: the text widget and the canvas widget. The text widget can be used to prototype text-based interfaces. Any character in the text can react to user input using tags. For example, it is possible to turn a string of characters into a hypertext link. In Beaudouin-Lafon (2000), the text widget was used to prototype a new method for finding and replacing text. When entering the search string, all occurrences of the string are highlighted in the text (Fig. 52.13). Once a replace string has been entered, clicking an occurrence replaces it (the highlighting changes from yellow to red). Clicking a replaced occurrence returns it to its original value. This example also uses the canvas widget to create a custom scrollbar that displays the positions and status of the occurrences.

The Tk canvas widget is a drawing surface that can contain arbitrary objects: lines, rectangles, ovals, polygons, text strings, and widgets. Tags allow behaviors (e.g., scripts) that are called when the user acts on these objects. For example, an object that can be dragged will be assigned a tag with three behaviors: button-press, mouse-move, and button-up. Because of the flexibility of the canvas, advanced visualization and interaction techniques can be implemented more quickly and easily than with other tools. For example, Fig. 52.14 shows a prototype exploring new ideas to manage overlapping windows on the screen (Beaudouin-Lafon, 2001). Windows can be stacked and slightly rotated so that it is easier to recognize them, and they can be folded so it is possible to see what is underneath without having to move the window. Even though the prototype is not perfect e.g., folding a window that contains text is not properly supported, it was instrumental in identifying a number of problems with the interaction techniques and finding appropriate solutions through iterative design.

Tcl and Tk can also be used with other programming languages. For example, Pad++ (Bederson & Meyer, 1998) is implemented as an extension to Tcl/Tk: the zoomable interface is implemented in C for performance and is accessible from Tk as a new widget. This makes it easy to prototype interfaces that use zooming. It is also a way to develop evolutionary prototypes: a first prototype is implemented completely in Tcl, then parts of it are reimplemented in a compiled language to performance. Ultimately, the complete system may be implemented in another language, although it is more likely that some parts will remain in Tcl.

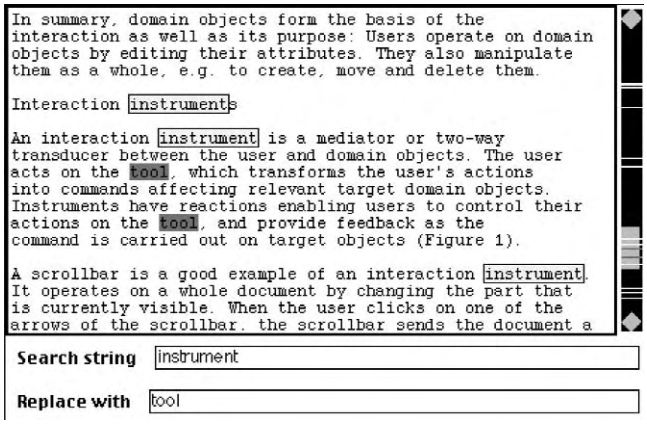


FIGURE 52.13. Using the Tk text and canvas widgets to prototype a novel search and replace interaction technique (Beaudouin-Lafon, 2000).

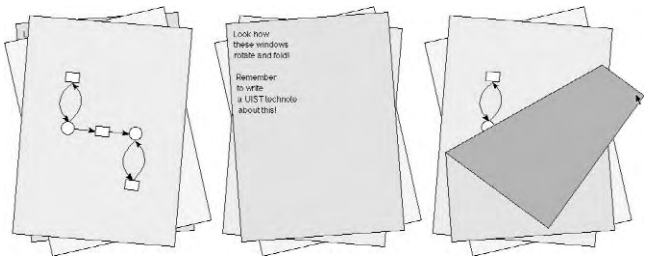


FIGURE 52.14. Using the Tk canvas widget to prototype a novel window manager (Beaudouin-Lafon, 2001).

Software prototypes can also be used in conjunction with hardware prototypes. Fig. 52.15 shows an example of a hardware prototype that captures handwritten text from a paper flight strip (using a combination of a graphics tablet and a custom-designed system for detecting the position of the paper strip holder). We used Tcl/Tk, in conjunction with C++, to present information on a RADAR screen (tied to an existing air traffic control simulator) and to provide feedback on a touch-sensitive display next to the paper flight strips (Caméléon, Mackay, et al., 1998). The user can write in the ordinary way on the paper flight strip, and the system interprets the gestures according to the location of the writing on the strip. For example, a change in flight level is automatically sent to another controller for confirmation and a physical tap on the strip's ID lights up the corresponding aircraft on the RADAR screen. A later section of this chapter expands this approach to the development of prototypes of mixed reality and pervasive computing systems.

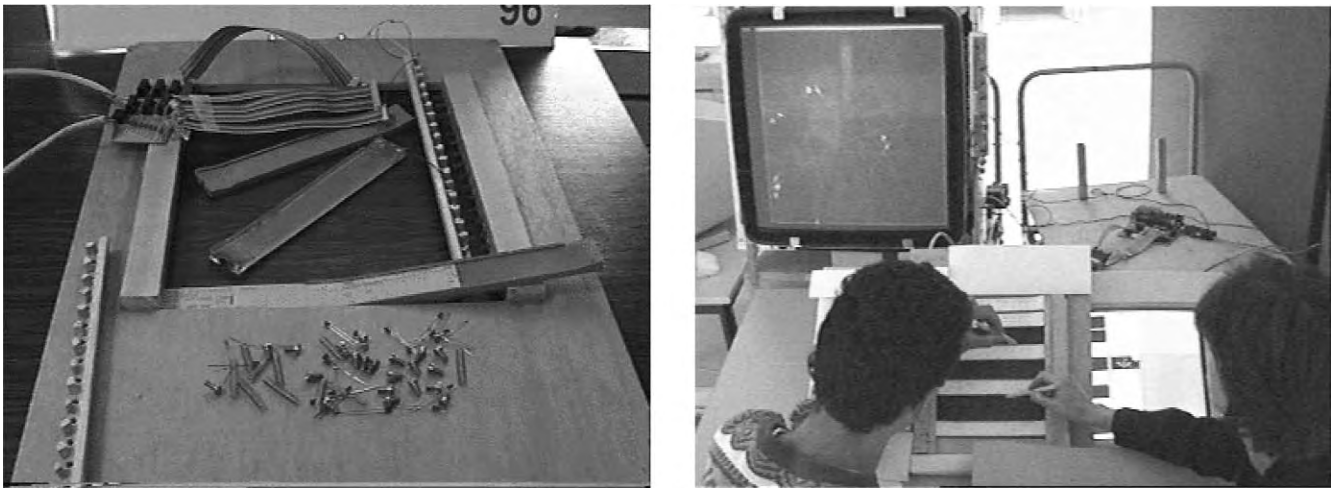


FIGURE 52.15. Caméléon's augmented stripboard (left) is a working hardware prototype that identifies and captures hand-writing from paper flight strips. Members of the design team test the system (right), which combines both hardware and software prototypes into a single interactive simulation.

ITERATIVE AND EVOLUTIONARY PROTOTYPES

Prototypes may also be developed with traditional software development tools. In particular, high-precision prototypes usually require a level of performance that cannot be achieved with the rapid online prototyping techniques previously described. Similarly, evolutionary prototypes intended to evolve into the final product require more traditional software development tools. Finally, even shipped products are not final, since subsequent releases can be viewed as initial designs for prototyping the next release.

Development tools for interactive systems have been in use for over 20 years and are constantly being refined. Several studies have shown that the part of the development cost of an application spent on the user interface is 50% to 80% of the total cost of the project (Myers & Rosson, 1992). The goal of development tools is to shift this balance by reducing production and maintenance costs. Another goal of development tools is to anticipate the evolution of the system over successive releases and support iterative design.

The lowest level tools are graphical libraries, which provide hardware independence for painting pixels on a screen and handling user input and window systems that provide an abstraction (the window) to structure the screen into several virtual terminals. User interface toolkits structure an interface as a tree of interactive objects called "widgets," while user interface builders provide an interactive application to create and edit those widget trees. Application frameworks build on toolkits and UI builders to facilitate creation of typical functions such as cut/copy/paste, undo, help and interfaces based on editing multiple documents in separate windows. Model-based tools semiautomatically derive an interface from a specification of the domain objects and functions to be supported. Finally, user interface development environments (UIDEs) provide an integrated collection of tools for the development of interactive software.

Before we describe some of these categories in more detail, it is important to understand how they can be used for prototyping. It is not always best to use the highest level available tool for prototyping. High-level tools are most valuable in the long term because they make it easier to maintain the system, port it to various platforms, or localize it to different languages. These issues are irrelevant for vertical and throwaway prototypes, so a high-level tool may prove less effective than a lower level one.

The main disadvantage of higher level tools is that they constrain or stereotype the types of interfaces they can implement. User interface toolkits usually contain a limited set of widgets, and it is expensive to create new ones. If the design must incorporate new interaction techniques, such as bimanual interaction (Kurtenbach, Fitzmaurice, Baudel, & Buxton, 1997) or zoomable interfaces (Bederson & Hollan, 1994), a user interface toolkit will hinder rather than help prototype development. Similarly, application frameworks assume a stereotyped application with a menu bar, several toolbars, a set of windows holding documents, and so forth. Such a framework would be inappropriate for developing a game or a multimedia educational CD-ROM that requires a fluid, dynamic, and original user interface.

Finally, developers need to truly master these tools, especially when prototyping in support of a design team. Success depends on the programmer's ability to change the details quickly as well as the overall structure of the prototype. A developer will be more productive when using a familiar tool than if forced to use a more powerful but unknown tool.

Since a complete tour of development tools for interactive systems is beyond the scope of this chapter, we focus on those tools that can be used most effectively for prototyping: user interface toolkits, user interface builders, and user interface development environments.

User Interface Toolkits

User interface toolkits are probably the most widely used tool nowadays to implement applications. All three major platforms (Unix/Linux, MacOS, and Windows) come with at least one standard UI toolkit. The main abstraction provided by a UI toolkit is the widget. A widget is a software object that has three facets that closely match the MVC model (Krasner & Pope, 1988): a presentation, a behavior, and an application interface.

The presentation defines the graphical aspect of the widget. The overall presentation of an interface is created by assembling widgets into a tree. Widgets such as buttons are the leaves of the tree. Composite widgets constitute the nodes of the tree and control the layout of their children. The behavior of a widget defines the interaction methods it supports: a button can be pressed, a scrollbar can be scrolled, and a text field can be edited. The application interface defines how a widget communicates the results of the user interaction to the rest of the application. It is usually based on a notification mechanism.

One limitation of widgets is that their behaviors are limited to the widget itself. Interaction techniques that involve multiple widgets, such as drag-and-drop, cannot be supported by the widgets' behaviors alone and require separate support in the UI toolkit. Some interaction techniques, such as toolglasses or magic lenses (Bier, Stone, Pier, Buxton, & De Rose, 1993), break the widget model both with respect to the presentation and the behavior and cannot be supported by traditional toolkits. In general, prototyping new interaction techniques requires either implementing them within new widget classes, which is not always possible, or not using a toolkit at all. Implementing a new widget class is typically more complicated than implementing the new technique outside the toolkit, for example, with a graphical library, and is rarely justified for prototyping. Many toolkits provide a blank widget, such as the Canvas in Tk or JFrame in Java Swing (Eckstein, Loy, & Wood, 1998), which can be used by the application to implement its own presentation and behavior. This is usually a good alternative to implementing a new widget class, even for production code.

User interface toolkits have been an active area of research over the past 15 years. InterViews (Linton, Vlissides, & Calder, 1989) inspired many modern toolkits and user interface builders. Some recent research toolkits that can be used for prototyping include SubArctic (Hudson, Mankoff, & Smith, 2005) and Satin (Hong & Landay, 2000). The latter is dedicated to ink-based interaction and is used with the Silk and Denim UIDEs described below.

A number of toolkits have also shifted away from the widget model to address other aspects of user interaction. For example, GroupKit (Roseman & Greenberg, 1996, 1999) was designed for groupware, Jazz (Bederson, Meyer, & Good, 2000) for zoomable interfaces, the Visualization (Schroeder, Martin, & Lorensen, 1997) and InfoVis (Fekete, 2004) toolkits for visualization, Inventor (Strass, 1993) for 3-D graphics, and Metisse (Chapuis & Roussel, 2005) for window management (Fig. 52.16).

Creating an application or a prototype with a UI toolkit requires solid knowledge of the toolkit and experience with programming interactive applications. In order to control the complexity of the interrelations between independent pieces of code (creation of widgets, callbacks, global variables, etc.), it is important to use well-known design patterns (Gamma, Helm, Johnson, & Vlissides, 1995) such as Command, Chain of responsibility, Mediator, and Observer. Otherwise the code quickly becomes unmanageable and, in the case of a prototype, unsuitable to design space exploration.

User Interface Builders

User interface builders leverage user interface toolkits by allowing the developer of an interactive system to create the presentation of the user interface, such as the tree of widgets, interactively with a graphical editor. The editor typically features a

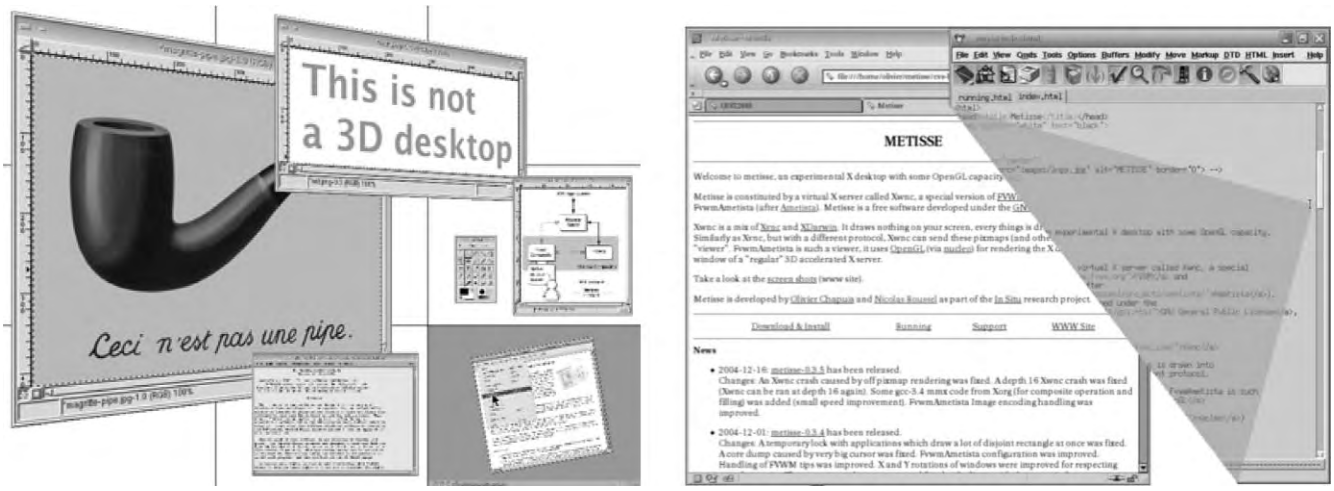


FIGURE 52.16. Two prototypes of a window system implemented with Metisse. The one on the right implements folding windows. Unlike in Figure 15 above, here it works with real applications. (© O. Chapuis, with permission)

palette of widgets that the user can use to draw the interface in the same way as a graphical editor is used to create diagrams with lines, circles, and rectangles. The presentation attributes of each widget can be edited interactively as well as the overall layout. This saves a lot of time that would otherwise be spent writing and fine tuning rather dull code that creates widgets and specifies their attributes. It also makes it extremely easy to explore and test design alternatives.

Apple's Interface Builder (Fig. 52.17) is a descendant of the NeXT interface builder (NeXT Corporation, 1991). The palette to the right contains the available widgets. The user interface of the application was created by dragging these widgets into the application window at the top left. The bottom left window contains icons representing the application objects. By dragging connectors between widgets and these objects, a significant part of the behavior of the interface can be created interactively. The user interface can be tested at any time by switching the builder to test mode, making it easy to verify that it behaves as expected. The same application built directly with the underlying toolkit would require dozens of lines of code and significant debugging.

User interface builders are widely used to develop prototypes, as well as final applications. They are easy to use, they make it easy to change the look of the interface, and they hide a lot of the complexity of creating user interfaces with UI toolkits. However, despite their name, they do not cover the whole user interface. Therefore they still require a significant amount of programming, a good knowledge of the underlying toolkit and an understanding of their limits, especially when prototyping novel visualization and interaction techniques.

User Interface Development Environments

A number of high-level tools exist for creating interactive applications. They are often based on user interface toolkits, and they sometimes include an interface builder. These tools are often referred to as "user interface development environments."

The simplest of these tools are application frameworks, which address stereotyped applications. For example, many applications have a standard form where windows represent documents that can be edited with menu commands and tools from palettes; each document may be saved into a disk file; standard functions such as copy/paste, undo, and help are supported. Implementing such stereotyped applications with a UI toolkit or UI builder requires replicating a significant amount of code to implement the general logic of the application and the basics of the standard functions. Application frameworks address this issue by providing a shell that the developer fills in with the functional core and the actual presentation of the nonstandard parts of the interface. Most frameworks have been inspired by MacApp, a framework developed in the 1980s to develop applications for the Macintosh (Apple Computer, 1996). Some frameworks are more specialized than MacApp. For example, Unidraw (Vlissides & Linton, 1990) is a framework for creating graphical editors in domains, such as technical and artistic drawing, music composition, or circuit design. By addressing a smaller set of applications, such a framework can provide more support and significantly reduce implementation time.

Mastering an application framework takes time. It requires knowledge of the underlying toolkit and the design patterns used in the framework, and a good understanding of the de-

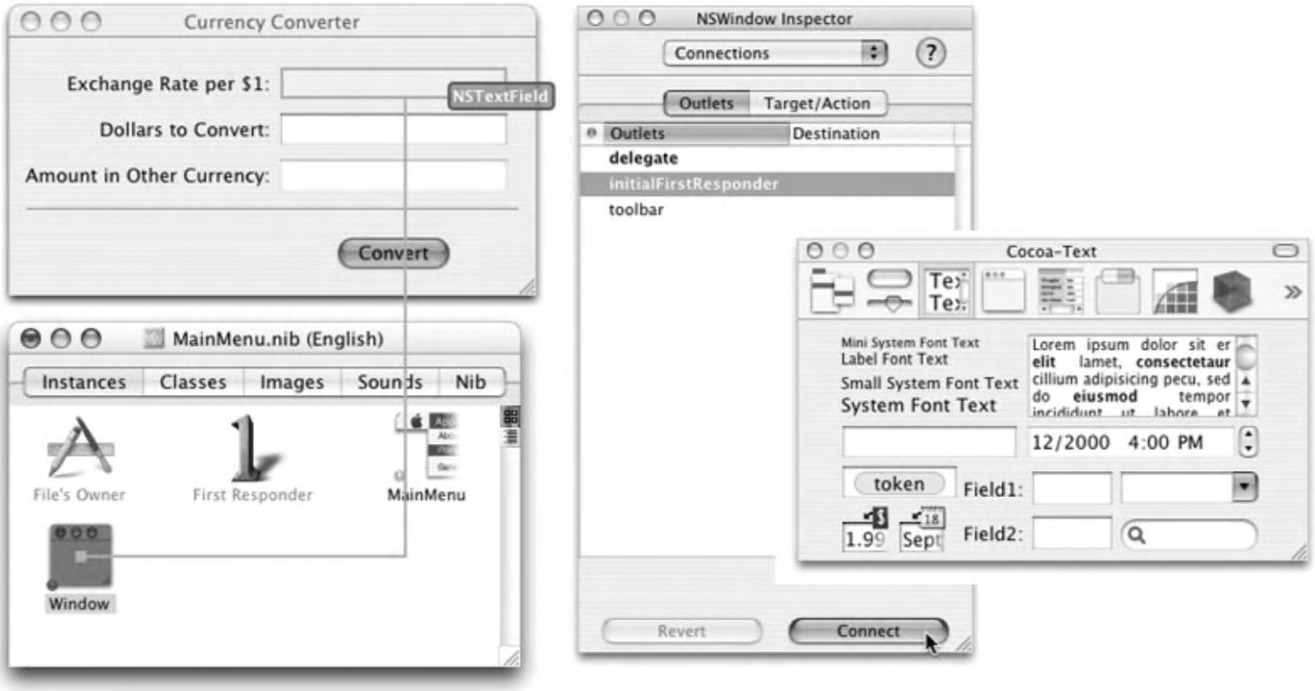


FIGURE 52.17. Interface Builder with the window being built (top-left), application objects (bottom-left), inspector (center) and widget palette (right).

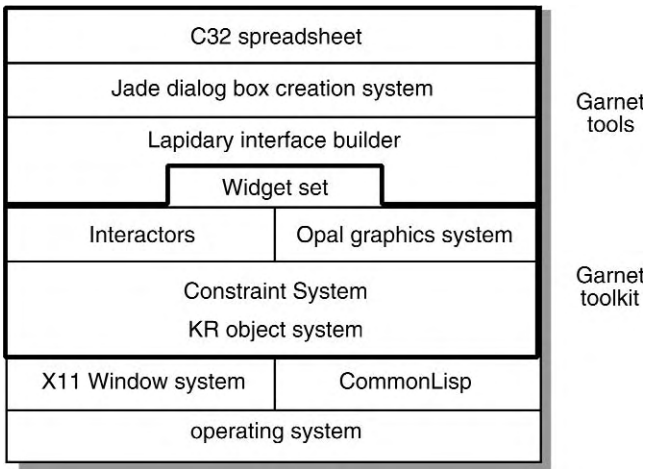


FIGURE 52.18. The Garnet toolkit and tools (Myers et al., 1990).

sign philosophy of the framework. A framework is useful because it provides a number of functions "for free," but at the same time it constrains the design space that can be explored. Frameworks can prove effective for prototyping if their limits are well understood by the design team.

UIDEs consist in assembling a set of tools into an environment where different aspects of an interactive system can be specified and generated separately. For example, Garnet (Fig. 52.18) and its successor Amulet (Myers, Giuse, et al., 1990; Myers, McDaniel, et al., 1997) provide a comprehensive set of tools, including a traditional user interface builder, a semiautomatic tool for generating dialogue boxes, a user interface builder based on a demonstration approach, and so forth.

Some UIDEs include tools that are specifically designed for prototyping. For example, Silk (Landay & Myers, 2001) is a tool aimed at the early stages of design, when interfaces are sketched rather than prototyped in software. The user sketches a user interface directly on the screen (Fig. 52.19). Using gesture

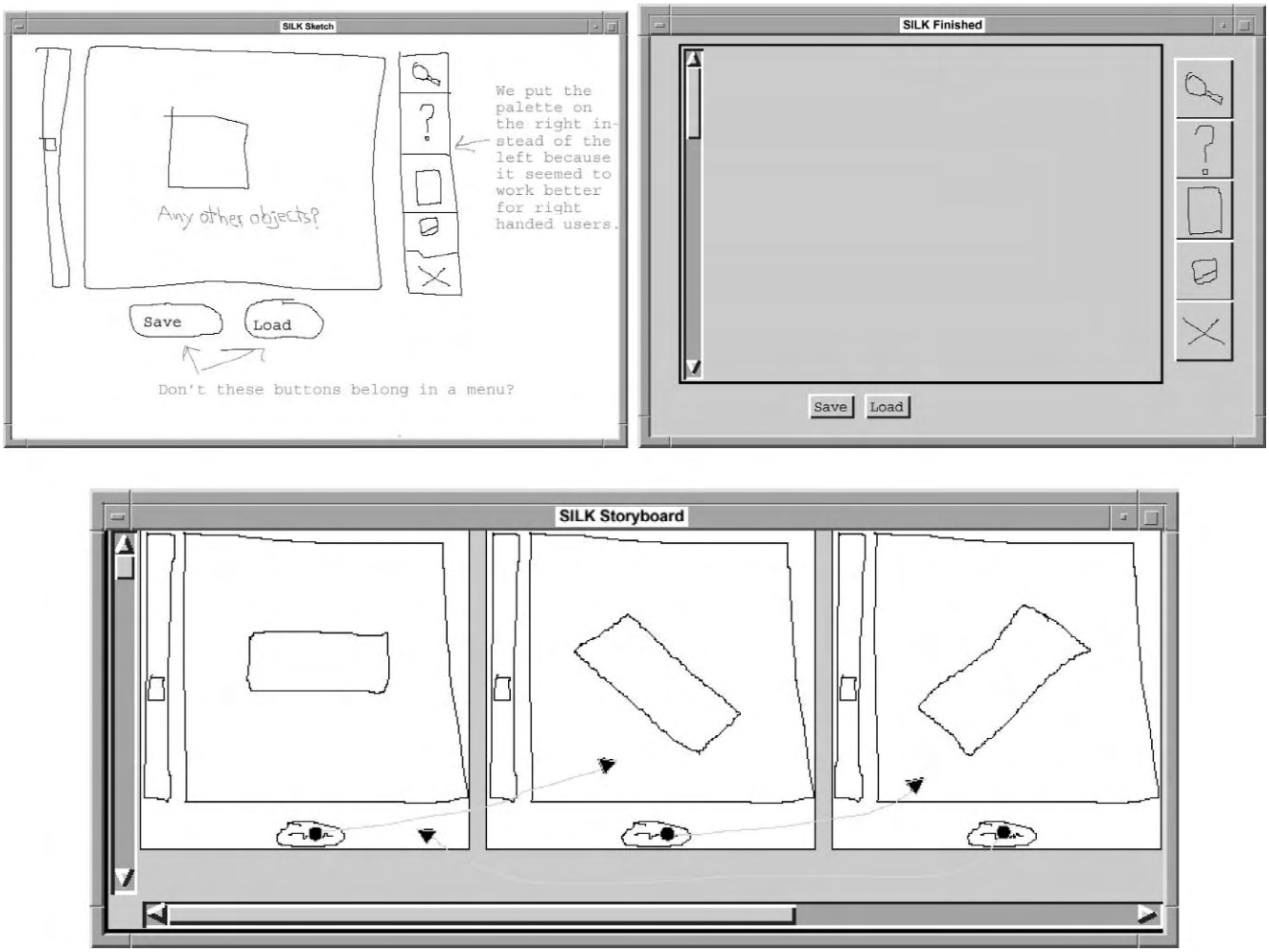


FIGURE 52.19. A sketch created with Silk (top left) and its automatic transformation into a Motif user interface (top right). A storyboard (bottom) used to test sequences of interactions, here a button that rotates an object. (© J. Landay, with permission)

recognition, Silk interprets the marks as widgets, annotations, and so forth. Even in its sketched form, the user interface is functional: buttons can be pressed, tools can be selected in a toolbar, and so on. The sketch can also be turned into an actual interface, i.e., using the Motif toolkit. Finally, storyboards can be created to describe and test sequences of interactions. Monet (Li & Landay, 2005) expands this approach by supporting the specification of animations and continuous interaction such as drag-and-drop. Silk and Monet therefore combine some aspects of offline and online prototyping techniques, trying to get the best of both worlds. Another example is Denim (Lin, Newman, Hong, & Landay, 2000), which addresses the prototyping of websites. Like Silk, it uses a sketch-based interface to allow designers to quickly enter ideas about the overall design of the website as well as individual pages and test the navigation. This illustrates a current trend in research where online tools attempt to support not only the development of the final system, but the whole design process. The following section will also illustrate this trend in the new and very active areas of mixed reality and pervasive computing.

PROTOTYPING MIXED REALITY AND PERVASIVE COMPUTING SYSTEMS

While most examples in the previous sections concerned traditional graphical user interfaces (GUIs), there is an increasing need to address the design and prototyping of systems that combine the physical and online worlds. The current trend towards pervasive computing and mixed reality started with Weiser's (1991) seminal article on ubiquitous computing and was carried on by augmented reality (Wellner, Mackay, & Gold, 1993) and tangible interfaces (Holmquist, Schmidt, & Ullmer, 2004). A common theme of these approaches is to combine interaction with the real world and interaction with online information, taking advantage of humans' abilities to interact with physical artifacts.

While mixed reality emphasizes the role of physical objects, pervasive computing emphasizes the role of the physical space. Both raise new and difficult issues when trying to actually design and build such systems. At the prototyping stage, they typically require a larger effort than when designing a GUI. First, the range of possible interactions is much broader and typically includes gesture, eye-gaze, or speech. These techniques are already difficult to incorporate into traditional interfaces: in a mixed reality or pervasive computing environment, the user is much less constrained, making recognition and interpretation of users' actions much harder. Designers must then address recognition errors and more generally the effects of context on the sensing techniques. Second, the range of artifacts to design is larger than with desktop interfaces, as it includes the physical artifacts that the users may manipulate; in pervasive computing, the role of user location, the issues of wireless network coverage and sensor range, the difficulty of providing feedback where needed are all new challenges that need to be addressed. Finally, these systems are not necessarily used for tasks where speed of execution and productivity are the primary measures of success. Whether they are used in the home, for tourism, such as tour guides, for assisted living or plain entertainment, the wider and less well-mapped design space typically require extensive prototyping work.

For example, Boucher and Gaver (2006) described their experiences in designing the drift table, a coffee table with a small screen in the center displaying an aerial view of England and load sensors measuring the weight of objects on the table. According to the measured weight and its estimated location, the aerial view slowly drifts in that direction, as if navigating with a hot air balloon. While the resulting design is conceptually simple, it required a many prototypes, both to explore what it should do and how it could be built. The final prototype (Fig. 52.20) had to be aesthetically pleasing as well as fully functional for testing in people's homes. This example shows that the higher precision prototypes had to address interaction, function, and looks simultaneously. This proved particularly challenging but is common when prototyping these types of systems.

The prototyping strategies that we have described, such as horizontal, vertical, and scenario-based prototypes, are still valid for prototyping mixed reality and pervasive computing systems, as well as some of the prototyping techniques, such as offline prototyping with mock ups and video prototyping. Online prototyping, however, requires specific tools and iterative and incremental prototyping, even more so. The rest of this section explores some of the existing tools.



FIGURE 52.20. The drift table (© W. Gaver, with permission)

Phidgets (Greenberg & Fitchett, 2001) are physical sensors and actuators that can be controlled from a computer through a USB connection. A wide range of sensors is available (light, motion, distance, pressure, humidity, etc.), as well as input devices (buttons, sliders), LEDs, LCD displays, motors, and RFID tag readers. By hiding their implementations and exposing their functionalities through a standard software interface, phidgets can greatly facilitate the design and prototyping of tangible interfaces. For example, they can be embedded into a physical artifact and programmed from the computer to prototype a mobile device. Since all phidgets can be represented on the computer screen, they can also be used for Wizard of Oz settings, where the Wizard can directly control the output phidgets such as LEDs and motors.

Papier-Mâché (Klemmer, Li, Lin, & Landay, 2004) is a toolkit for creating tangible interfaces based on vision, RFID tags, and barcodes. It is more adapted to the development of augmented paper applications than Phidgets and provides a richer development environment. The DART toolkit (MacIntyre, Gandy, Dow, & Bolter, 2004) takes a similar approach, but targets augmented and mixed reality applications. The development environment is based on Macromedia Director, with the explicit goal of being familiar to designers. In the context of prototyping, both toolkits can be used for applications other than those that they explicitly target. For example, DART can superimpose a computer-generated model with live or recorded video, which can be used to put an image or 3-D model of a device being designed in the real world.

Topiary (Li, Hong, & Landay, 2004) is a tool for prototyping pervasive applications that use the locations of people, places, and objects. Unlike the above tool, its goal is not to create real-world prototypes allowing to test a system in situ. Rather, it is an online tool that allows to create and test scenarios on the screen, using maps representing the environment and icons representing the people, places, and objects of interest. Once a scenario has been created, the tool allows the user to experience it either from a bird's eye view or from the perspective of a particular user by displaying the user's PDA. Such scenario-based prototyping can be very useful in the early stages of design, when exploring ideas.

Finally, a CAPpella (Dey, Hamid, Beckmann, Li, & Hsu, 2004), is a tool to help end users create or customize context-aware applications. It uses programming by demonstration so that users only have to show examples of sensor data and the system's reaction. While the system targets end users, it can also be used by designers to create prototypes of context-aware applications. By hiding the complexity of programming recognizers, it allows for a fast exploration of alternatives.

In summary, while mixed reality and pervasive computing create new challenges for designers, a number of tools are starting to appear to help with the design process in general and prototyping in particular. Interestingly, these tools can also be used to prototype more traditional applications. For example, phidgets can be used to control an online application in a Wizard of Oz setting, DART can be used to create scenarios that mix videos with models of future artifacts, and a CAPpella can be used to experiment with recognition-based applications. As mixed reality and pervasive computing systems become more widely available, they will no doubt provide more tools to be used by designers in unexpected ways to help with the prototyping of interactive systems at large.

CONCLUSION

Prototyping is an essential component of interactive system design. Prototypes may take many forms, from rough sketches to detailed working prototypes. They provide concrete representations of design ideas and give designers, users, developers and managers an early glimpse into how the new system will look and feel. Prototypes increase creativity, allow early evaluation of design ideas, help designers think through and solve design problems, and support communication within multidisciplinary design teams.

Prototypes, because they are concrete and not abstract, provide a rich medium for exploring a design space. They suggest alternate design paths and reveal important details about particular design decisions. They force designers to be creative and to articulate their design decisions. Prototypes embody design ideas and encourage designers to confront their differences of opinion. The precise aspects of a prototype offer specific design solutions: designers can then decide to generate and compare alternatives. The imprecise or incomplete aspects of a prototype highlight the areas that must be refined or require additional ideas.

We defined prototypes and then discussed them as design artifacts. We introduced four dimensions by which they can be analyzed: representation, precision, interactivity, and evolution. We then discussed the role of prototyping within the design process and explained the concept of creating, exploring, and modifying a design space. We briefly described techniques for generating new ideas to expand the design space and techniques for choosing among design alternatives to contract the design space.

We described a variety of rapid prototyping techniques for exploring ideas quickly and inexpensively in the early stages of design, including offline techniques (from paper and pencil to video) and online techniques (from fixed to interactive simulations). We then described iterative and evolutionary prototyping techniques for working out the details of the online interaction, including development tools and software environments. Finally, we addressed the emerging fields of mixed reality and pervasive computing and described the new challenges they raise for interactive systems design, as well as some of the tools available for prototyping them.

We view design as an active process of working with a design space, expanding it by generating new ideas and contracting as design choices are made. Prototypes are flexible tools that help designers envision this design space, reflect upon it, and test their design decisions. Prototypes are diverse and can fit within any part of the design process, from the earliest ideas to the final details of the design. Perhaps most important, prototypes provide one of the most effective means for designers to communicate with each other, as well as with users, developers, and managers, throughout the design process.

References

Apple Computer. (1996). *Programmer's guide to MacApp.* Cupertino, CA: Apple Computer, Inc.

Beaudouin-Lafon, M. (2000). Instrumental interaction: An interaction model for designing post-WIMP user interfaces. *Proceedings ACM Human Factors in Computing Systems, CHI'2000*, The Hague, Netherlands, 446–453.

Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping Windows. *Proceedings of ACM Symposium on User Interface Software and Technology, UIST 2001*, Orlando, Florida. *CHI Letters*, 3(2), 153–154.

Beaudouin-Lafon, M. (2004, May). Designing interaction, not interfaces. *Proceedings of the Working Conference on Advanced Visual interfaces, AVI'04* (pp. 15–22). Gallipoli, Italy.

Beaudouin-Lafon, M., & Lassen, M. (2000). The architecture and implementation of a post-WIMP graphical application. *Proceedings of ACM Symposium on User Interface Software and Technology, UIST 2000,* San Diego, CA. *CHI Letters 2*(2), 191–190.

Beaudouin-Lafon, M., & Mackay, W. (2000, May). Reification, polymorphism and reuse: Three principles for designing visual interfaces. *Proceedings Conference on Advanced Visual Interfaces, AVI 2000*, (pp. 102–109). Palermo, Italy.

Beck, K. (2000). *Extreme programing explained*. New York: Addison-Wesley.

Bederson, B., & Hollan, J. (1994). Pad++: A zooming graphical interface for exploring alternate interface physics. *Proceedings of ACM Symposium on User Interface Software and Technology, UIST'94*, Marina del Rey, 17–26.

Bederson, B., & Meyer, J. (1998). Implementing a zooming interface: Experience building Pad++. *Software Practice and Experience, 28*(10), 1101–1135.

Bederson, B. B., Meyer, J., & Good, L. (2000). Jazz: An extensible zoomable user interface graphics toolkit in Java. *Proceedings of ACM Symposium on User Interface Software and Technology, UIST 2000,* San Diego, CA. *CHI Letters 2*(2), 171–180.

Berger, N. (2006). The excel story. *Interactions, 13*(1),14–17.

Bier, E., Stone, M., Pier, K., Buxton, W., & De Rose, T. (1993). Toolglass and magic lenses: The see-through interface. *Proceedings ACM SIGGRAPH* (pp. 73–80). Anaheim, CA.

Bødker, S., (1999). Scenarios in user-centered design: Setting the stage for reflection and action. *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, HICSS-32* (Vol 3, article 3053, 11 pages). Wailea, HI: IEEE Computer Society.

Bødker, S., Ehn, P., Knudsen, J., Kyng, M., & Madsen, K. (1988). Computer support for cooperative design. *Proceedings of the CSCW'88 ACM Conference on Computer-Supported Cooperative Work, (pp.* 377–393). Portland, OR.

Boehm, B. (1988). A spiral model of software development and enhancement. *IEEE Computer, 21*(5), 61–72.

Boucher, A., & Gaver, W. (2006). Developing the drift table. *Interactions, 13*(1), 24–27.

Chapanis, A. (1982). Man/Computer research at Johns Hopkins. In R. A. Kasschau, R. Lachman, & K. R. Laughery (Eds.), *Information technology and psychology: Prospects for the future* (238–249). New York, NY: Praeger Publishers, Third Houston Symposium.

Chapuis, O., & Roussel, N. (2005, October). Metisse is not a 3D desktop! *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology, UIST '05* (pp. 13–22). Seattle, WA.

Collaros, P. A., & Anderson, L. R. (1969). Effect of perceived expertness upon creativity of members of brainstorming groups. *Journal of Applied Psychology, 53*, 159–163.

de Vreede, G.-J., Briggs, R., van Duin, R., & Enserink, B. (2000). Athletics in electronic brainstorming: Asynchronous brainstorming in very large groups. *Proceedings of the 33nd Annual Hawaii International Conference on System Sciences, HICSS-33* (Vol 1, article 1042, 11 pages). Wailea, HI: IEEE Computer Society.

Dey, A. K., Hamid, R., Beckmann, C., Li, I., & Hsu, D. (2004, April). A CAPpella: Programming by demonstration of context-aware applications. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04* (pp. 33–40). Vienna, Austria.

Diehl, M., & Stroebe, W. (1987). Productivity loss in brainstorming groups: Towards the solution of a riddle. *Journal of Personality and Social Psychology, 53*(3), 497–509. Washington, D.C.: American Psycology Association.

Dijkstra-Erikson, E., Mackay, W. E., & Arnowitz, J. (2001, March). Trialogue on design of. *ACM/Interactions,* 109–117.

Dourish, P. (1997). Accounting for system behaviour: Representation, reflection and resourceful action. In M. Kyng, & L. Mathiassen, (Eds.), *Computers and design in context* (pp. 145–170). Cambridge, MA: MIT Press.

Eckstein, R., Loy, M., & Wood, D. (1998). *Java swing*. Cambridge, MA: O'Reilly.

Fekete, J.-D. (2004). The InfoVis toolkit. *Proceedings of the 10th IEEE Symposium on Information Visualization, InfoVis'04, IEEE Press* (pp. 167–174) Austin, Texas.

Frishberg, N. (2006). Prototyping with junk. *Interactions, 13*(1), 21–23.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns, elements of reusable object-oriented software*. Reading, MA: Addison-Wesley.

Good, M. D., Whiteside, J. A., Wixon, D. R., & Jones, S. J. (1984). Building a user-derived interface. *Communications of the ACM, 27*(10), 1032–1043. New York: ACM.

Goodman, D. (1987). *The complete HyperCard handbook*. New York, NY: Bantam Books.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Greenberg, S., & Fitchett, C. (2001, November). Phidgets: Easy development of physical interfaces through physical widgets. *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology, UIST '01* (pp. 209–218). Orlando, FL.

Holmquist, L. E., Schmidt, A., & Ullmer, B. (2004). Tangible interfaces in perspective. *Personal and Ubiquitous Computing, 8*(5), 291–293. Heidelberg: Springer.

Hong, J. I., & Landay, J. A. (2000, November). SATIN: A toolkit for informal ink-based applications. *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology, UIST '00* (pp. 63–72). San Diego, CA.

Houde, S., & Hill, C. (1997). What do prototypes prototype? In M. G. Helaner, T. K. Landauer. & P. V. Pradhu (Eds.) *Handbook of human computer interaction* (2nd ed.) (pp. 367–381). North-Holland

Hudson, S. E., Mankoff, J., & Smith, I. (2005, April). Extensible input handling in the subArctic toolkit. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '05* (pp. 381–390). Portland, Oregon.

Kelley, J. F. (1983). An empirical methodology for writing user-friendly natural language computer applications. *Proceedings of CHI '83 Conference on Human Factors in Computing Systems.* Boston, MA, 193–196.

Klemmer, S. R., Li, J., Lin, J., & Landay, J. A. (2004, April). Papier-mache: Toolkit support for tangible input. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04* (pp. 399–406). Vienne, Austria.

Krasner, E. G., & Pope, S. T. (1988, August/September). A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming*, 27–49.

Kurtenbach, G., Fitzmaurice, G., Baudel, T., & Buxton. W. (1993). The design of a GUI paradigm based on tablets, two-hands, and transparency. *Proceedings of ACM Human Factors in Computing Systems, CHI'97* (pp. 35–42). Atlanta, GA.

Landay, J., & Myers, B. A. (2001). Sketching interfaces: Toward more human interface design. *IEEE Computer, 34*(3), 56–64.

Li, Y., & Landay, J. A. (2005, October). Informal prototyping of continuous graphical interactions by demonstration. *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology, UIST '05* (pp. 221–230). Seattle, WA.

Li, Y., Hong, J. I., & Landay, J. A. (2004, October). Topiary: A tool for prototyping location-enhanced applications. *Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology, UIST '04* (pp. 217–226). Santa Fe, NM.

Lin, J., Newman, M. W., Hong, J. I., & Landay, J. A. (2000, April). DENIM: Finding a tighter fit between tools and practice for website design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00* (pp. 510–517). Hague, Netherlands.

Linton, M. A., Vlissides, J. M., & Calder, P. R. (1989). Composing user interfaces with InterViews. *IEEE Computer, 22*(2), 8–22.

MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2005). DART: A toolkit for rapid design exploration of augmented reality experiences. *ACM Transactions on Graphics, 24*(3), 932–932.

Mackay, W.E. (1988). Video prototyping: A technique for developing hypermedia systems. *Conference Companion of ACM CHI'88, Conference on Human Factors in Computing. Washington DC.* Retrieved April 2, 2007, from http://www.lri.fr/~mackay/publications.html.

Mackay, W. E. (1995). Ethics, lies and videotape. *Proceedings ACM Human Factors in Computing Systems, CHI'95* (pp. 138–145). Denver, CO.

Mackay, W. E. (2000). Video techniques for participatory design: Observation, brainstorming & prototyping. *Tutorial Notes, CHI 2000, Human Factors in Computing Systems,* Hague, Netherlands. Retrieved April 2, 2007 from http://www.lri.fr/~mackay/publications.html.

Mackay, W.E. (2002). Using video to support interaction design, DVD tutorial, INRIA and ACM/SIGCHI. Retrieved April 2, 2007 from http://stream.cc.gt.atl.ga.us/hccvideos/viddesign.php

Mackay, W. E., & Fayard, A.-L. (1997). HCI, natural science and design: A framework for triangulation across disciplines. *Proceedings of ACM DIS '97, Designing Interactive Systems* (pp. 223–234). Pays-Bas, Amsterdam.

Mackay, W. E., & Pagani, D. (1994). Video mosaic: Laying out time in a physical space. *Proceedings of ACM Multimedia '94* (pp. 165–172). San Francisco, CA.

Mackay, W., Fayard, A. L., Frobert, L., & Médini, L. (1998) Reinventing the familiar: Exploring an augmented reality design space for air traffic Control. *Proceedings of ACM CHI '98 Human Factors in computing Systems* (pp. 558–565). Los Angeles, CA.

Mackay, W., Ratzer, A., & Janecek, P. (2000). Video artifacts for design: Bridging the gap between abstraction and detail. *Proceedings ACM Conference on Designing Interactive Systems, DIS 2000* (pp. 72–82). New York, NY.

Muller, M. J. (1991). PICTIVE: An exploration in participatory design. *Proceedigs of ACM CHI'91 Human Factors in Computing Systems* (pp. 225–231). New Orleans, LA.

Myers, B. A., & Rosson, M. B. (1992). Survey on user interface programming. *ACM Conference on Human Factors in Computing Systems, CHI'92* (pp. 195–202). Monterey, CA.

Myers, B. A., Giuse, D. A., Dannenberg, R. B., Vander Zander, B., Kosbie, D. S., Pervin, E., et al. (1990). Garnet: Comprehensive support for graphical, highly-interactive user interfaces. *IEEE Computer, 23*(11), 71–85.

Myers, B. A., McDaniel, R. G., Miller, R. C., Ferrency, A. S., Faulring, A., Kyle, B. D., et al. (1997). The Amulet environment. *IEEE Transactions on Software Engineering, 23*(6), 347–365.

NeXT Corporation. (1991). *NeXT interface builder reference manual*. Redwood City, CA: NeXT Corporation.

Norman, D. A., & Draper S. W. (Eds.). (1986). *User centered system design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Osborn, A. (1957). *Applied imagination: Principles and procedures of creative thinking* (Rev. ed.). New York, NY: Scribner's.

Ousterhout, J. K. (1994). *Tcl and the Tk toolkit*. Reading, MA: Addison Wesley.

Perkins, R., Keller, D. S., & Ludolph, F (1997). Inventing the Lisa user interface. *ACM Interactions, 4*(1), 40–53.

Rettig, M. (1994). Prototyping for tiny fingers. *Communication ACM, 37*(4), 21–27.

Raskin, J. (2000). *The humane interface*. New York, NY: Addison-Wesley.

Roseman, M., & Greenberg, S. (1996). Building real-time groupware with GroupKit, a groupware toolkit. *ACM Transactions on Computer-Human Interaction, 3*(1), 66–106.

Roseman, M., & Greenberg, S. (1999). Groupware toolkits for synchronous work. In M. Beaudouin-Lafon (Ed.), *Computer-supported cooperative work*: *Trends in software series* (pp. 135–168). Chichester: Wiley.

Schroeder, W., Martin, K., & Lorensen, B. (1997). *The visualization toolkit*. Upper Saddle River, NJ: Prentice Hall.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. San Francisco, CA: Morgan Kaufmann.

Strass, P. (1993). IRIS inventor, a 3D graphics toolkit. *Proceedings ACM Conference on Object-Oriented Programming, Systems, Languages and Applications*, *OOPSLA'93* (pp. 192–200). Washington, DC.

Vlissides, J. M., & Linton, M. A. (1990). Unidraw: A framework for building domain-specific graphical editors. *ACM Transactions on Information Systems, 8*(3), 237–268.

Weiser, M. (1991) The computer for the 21st century, *Scientific American, 265*(3), 94–104.

Wellner, P., Mackay, W. E. & Gold, R., (Eds.), (July 1993). Special Issue on Computer-Augmented Environments. *Communications of the ACM*. New York: ACM.

·53·

SCENARIO-BASED DESIGN

Mary Beth Rosson
John M. Carroll
The Pennsylvania State University

Introduction **1042**
Why Scenario-Based Design? **1042**
Scenarios are Concrete but Rough 1043
Scenarios Maintain an Orientation to People and Their Needs 1043
Scenarios are Evocative, Raising Questions at Many Levels 1044
Potential Pitfalls in Scenario-Based Design 1045
Scenario-Based Usability Engineering **1045**
Problem Analysis 1046
Activity Design 1048
User-Interaction Design 1050
Usability Evaluation 1052
Scenario-Based Iterative Design 1054
Applications of Scenario-Based Design **1054**
Scenarios in the System-Development Lifecycle 1054
Personas and envisionment. 1054
Use cases and object-oriented design 1055
Analyzing nonfunctional requirements 1055
Evaluating systems in use 1055
Scenario-Based Design Rationale as Theory 1056
Case-Based Learning about Scenario-Based Design . . . 1056
Current Challenges **1056**
References **1057**

INTRODUCTION

Scenario-based design is a family of techniques in which the use of a future system is concretely described at an early point in the development process. Narrative descriptions of envisioned usage episodes—user interaction scenarios—are then employed in a variety of ways to guide the development of the system that will enable these use experiences.

Like other user-centered approaches, scenario-based design changes the focus of design work from defining system operations (e.g., functional specification) to describing how people will use a system to accomplish work tasks and other activities. However, unlike approaches that consider human behavior and experience through formal analysis and modeling of well-specified tasks, scenario-based design offers relatively lightweight methods for envisioning future use possibilities.

A user interaction scenario is a "sketch of use." It is intended to vividly capture the essence of an interaction design, much as a two-dimensional paper-and-pencil sketch captures the essence of a physical design. Like any story, a scenario consists of a setting, or situation state; one or more actors with personal motivations; knowledge; and capabilities, and various tools and objects that the actors encounter and manipulate. The narrative describes a sequence of actions and events that lead to an outcome. These actions and events are related in a usage context that includes the goals, plans, and reactions of the people taking part in the episode.

Table 53.1 presents two brief scenarios in which Sharon, a high-school junior, attends an online science fair at which some of her friends are exhibiting their projects. In all of these scenarios, Sharon's goal is to chat with her friends and visit their exhibits. The scenarios contrast two ways that such a goal might be supported by computer network technologies. Each is a potential "solution" to Sharon's goals, but the user experience varies from asynchronous text-based reading and posting to a forum, to a real-time graphical simulation of a science fair that is in progress.

Designers can quickly construct scenarios like these in order to make envisioned possibilities more concrete. The example contrasts two contemporary approaches to online interactions, but not as an abstraction or a list of features or functions. It contrasts two episodes of human-computer interaction and personal experience.

User interaction scenarios can be successively detailed to discover and address finer-grained design issues. They serve as grist for group brainstorming, developing further alternatives, or raising questions about the assumptions behind the scenarios. They can be used to analyze software requirements, as a partial specification of functionality, and to guide the design of user interface layouts and controls. They can also be used to identify and plan evaluation tasks for usability tests.

TABLE 53.1. Two Contrasting Scenarios for a High-School Student Attending an Online Science Fair

A. Online Science Fair in a Web Forum
Sharon's friends Alicia and Delia have submitted projects to the online science fair, and they are counting on her to visit and talk to them while they "host" their exhibits tonight. At 7:30 P.M., she clicks on the URL in the e-mailed announcement to open a web browser on the science fair's home page. She sees a welcome notice and a categorical listing of projects at the fair (biology, forestry, astronomy, etc.). She knows that Alicia's project is on black holes, so she quickly scans the astronomy list to find her project and follows the link. She goes to another web page and admires Alicia's overview poster. She sees a number of links leading to the normal parts of a science project (problem, significance, methods, etc.). She also sees a link named "Talk to Me" and follows this, hoping to get in touch with Alicia. She sees a list of comments about the exhibits, from visitors as well as responses from Alicia. Alicia's most recent response was at 7:29 P.M., so Sharon thinks she may still be online. She posts a message ("Very flashy poster!") and is happy to get an immediate response from Alicia asking what else she has seen at the fair.
B. Online Science Fair in a Collaborative Virtual Environment
Sharon's friends Alicia and Delia have submitted projects to the online science fair, and they are counting on her to visit and talk to them while they "host" their exhibits tonight. At 7:30 P.M., she clicks on the URL in the e-mailed announcement, which launches a collaborative virtual environment. The welcome screen shows a two-dimension view of the middle-school gymnasium (she recognizes her favorite bleachers). She sees a bunch of little poster icons grouped in different areas of the gym, with banners announcing different science categories (biology, forestry, astronomy, etc.). She also sees a bunch of little faces moving around and recognizes a couple of her other friends, Brian and Corey. As she slides her cursor over the little poster icons, the project titles and authors are displayed, along with an invitation to zoom in. She knows that Alicia's project is on black holes, so she moves to the astronomy grouping, finds Alicia's project, and zooms in. She finds herself looking at a virtual poster, with a bunch of poster sections that name the traditional project components (problem, significance, methods, etc); there is also a comment board. She also sees an icon with Alicia's face "standing" nearby. When she double-clicks on Alicia, a text chat automatically opens, and she types a quick "Hello." Alicia replies and Sharon starts by teasing her about whether Brian has come to visit.

WHY SCENARIO-BASED DESIGN?

One reason that scenarios have become so popular in interactive system design is that they enable rapid communication about usage possibilities and concerns among many different stakeholders. It is easy to write a simple scenario such as those in Table 53.1, and takes only a little more effort to enrich it with a rough sketch or storyboard. When designers are working through ideas, they want to make progress quickly, so that they can obtain feedback and continue to refine their ideas. Scenarios are one way to do this.

The design of an interactive system is an ill-defined problem. Such problems tend to evoke a problem-solving strategy termed *solution-first* (Cross, 2001). In the solution-first strategy, designers generate and analyze a candidate solution as a means of clarifying the problem state, the allowable moves, and the goal. They exploit the concreteness of their own solution proposals to evoke further requirements for analysis.

A solution-first approach to design is energizing, effective, and efficient; it explains the popularity of contemporary system development approaches like rapid prototyping (Wasserman & Shewmaker, 1982) and extreme programming (Beck, 1999). But

this general strategy also entrains well-known hazards (Cross, 2001)—designers tend to generate solutions too quickly, before they analyze what is already known about the problem and possible moves. Once an approach is envisioned, they may have trouble abandoning it when it is no longer appropriate. Designers may too readily reuse pieces of a solution they have used earlier, pieces that are familiar and accessible, but perhaps not appropriate. They may not analyze their own solutions very well, or they may consider too few alternatives when exploring the problem space. In the next three sections we consider how scenario-based design may help to minimize these hazards of solution-first problem solving (see Table 53.2).

Scenarios are Concrete but Rough

Design analysis is always indeterminate, because the act of design changes the world within which people act and experience. Requirements always change (Brooks, 1995). When designs incorporate technologies that are evolving rapidly, requirements change even more rapidly. The more successful, the more widely adopted, and the more impact a design has, the less possible it will be to determine its correct design requirements. And in any case, refinements in software technology and new perceived opportunities and requirements propel a new generation of designs every two to three years.

Design representations that are at once concrete but flexible help to manage ambiguous and dynamic situations. Analysts must be concrete to avoid being swallowed by indeterminacy; they must be flexible to avoid being captured by false steps. Systematic decomposition is a traditional approach to managing ambiguity, but it does not promote flexibility. Instead, designers end up with a set of concrete subsolutions, each of which is fully specified. Unfortunately, by the time the set of subsolutions is specified, the requirements often have changed.

User interaction scenarios reconcile concreteness and flexibility. A scenario envisions a concrete design solution, but it can be couched at many levels of detail. Initially a scenario may be extremely rough. It specifies a possible design by indicating the tasks users may carry out, but without committing to lower-level details describing how the tasks will be carried out, or how the system will present the functionality for the tasks. The examples in Table 53.1 are at an intermediate level, with some information regarding task flow, but few details about specific user-system interactions.

Concrete material is interpreted more easily and more thoroughly than abstract material. For example, people remember a prototypical example far better than they remember the abstract category to which it belongs (Medin & Schaffer, 1978; Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). Incomplete material tends to be elaborated with respect to personal knowledge when it is encountered. This process of elaboration creates more robust and accessible memories, relative to memories for more complete material (Wertheimer, 1938). The combination of concreteness and incompleteness in scenarios engages a powerful variety of constructive cognitive processes.

The fluidity of design situations demands provisional solutions and tentative commitments. Yet if every design decision is suspended, the result will be a design space, not a design. A scenario is a concrete design proposal that a designer can evaluate and refine, but it is also rough, so that it can be easily altered, and many details can be deferred.

Scenarios Maintain an Orientation to People and Their Needs

Designers need constraints; too many things might be designed. The current state of technology development makes some solutions impossible and others irresistible. On the one hand, designers cannot use technology that does not yet exist. On the other hand, designers are caught up in a technological zeitgeist that biases them toward making use of the latest gadgets and gizmos. They are likely to be biased toward familiar technologies, even when they are aware of limitations in these technologies.

Scenarios are work-oriented design objects. They describe systems in terms of the work that users will try to do when they use those systems, ensuring that design will remain focused on the needs and concerns of users (Carroll & Rosson, 1990). Scenarios address what has been called the "representational bias" in human cognition: people overestimate the relevance of things that are familiar to them (Kahneman & Tversky 1972; Tversky & Kahneman, 1974). For instance, designers in a web-development company with years of experience in forms-based transactions will see this interaction paradigm as a solution to problems that might be better served by real-time interaction techniques. It is difficult to move beyond the familiar, but generating and sharing a vivid representation of exceptions to the status quo can promote innovating thinking. Scenarios describing unusual but critical circumstances can provide such a perspective.

The reuse of familiar ideas is just one type of constraint that designers may apply in their solution-first problem solving. Other constraints may arise from the organizational structures

TABLE 53.2. Concerns Stemming from She Solution-First Approach to Design, and Aspects of Scenario-Based Design that Address Each Concern

Hazards of the Solution-First Approach	How Scenario-Based Design Can Help
Designers want to select a solution approach quickly, which may lead to premature commitment to their first design ideas.	Because they are concrete but rough, scenarios support visible progress, but also relax commitment to the ideas expressed in the scenarios.
Designers attempt to quickly simplify the problem space with external constraints, such as the reusing of familiar solutions.	Because they emphasize people and their experiences, scenarios direct attention to the use-appropriateness of design ideas.
Designers are intent on elaborating their current design proposal, resulting in inadequate analysis of other ideas or alternatives.	Because they are evocative and by nature incomplete, scenarios promote empathy and raise usage questions at many levels.

within which the design work is embedded. Design projects are often chartered with a priori commitments to follow a systematic decomposition process. This makes them easy to manage, but unlikely to succeed with respect discovering the real requirements of users and clients. Schedules and resources are often assigned in ways that create ongoing conflicts between system designers and usability engineers. Usability engineers need to evaluate scenarios and prototypes at every stage of system development, but if schedules and resources do not provide for this, this work can conflict with software construction and refinement.

Constraints such as these can distract designers with ancillary factors so that they lose sight of what is essential in the design project—namely, the needs and concerns of users. The designer can become "unsituated" with respect to the real design situation, which is not the marketing manager's projections, or the instructional designer's list of steps, or the software engineer's system decomposition. The real design situation is the situation that will be experienced by the user, and designers need to stay focused on that.

Scenarios can be made even more effective as work-oriented design objects when users are directly involved in creating them. Ackoff (1979) argued that the indeterminacy of design situations makes it imperative that all stakeholders participate directly. Scenarios support a simple and natural process of participatory design, in which prospective users begin by enacting or relating episodes of current activities, then work iteratively with designers to transform and enrich these scenarios with the opportunities provided by new technologies (Carroll, Chin, Rosson, & Neale, 2000; Chin, Rosson, & Carroll, 1997).

Scenarios are Evocative, Raising Questions at Many Levels

There is a fundamental tension between thinking and doing: thinking impedes progress in doing, and doing obstructs thinking. Sometimes this conflict is quite sharp, as when one must stop and think before taking another step. But frequently it is more a matter of trading priorities. Designers are intelligent people performing complex and open-ended tasks. They want to reflect on their activities, and they routinely do reflect on their activities. However, people take pride not only in what they know and learn, but also in what they can do and in what they actually produce.

Donald Schön (1983; 1987) discussed this conflict extensively in his books on reflective practice. For example, he analyzed a coach reacting to an architecture student's design concept for a school building, which included a spiral ramp intended to maintain openness while breaking up lines of sight (she calls the idea "a Guggenheim"). Schön wrote that "when I visited open schools, the one thing they complained about was the warehouse quality—of being able to see for miles. It [the ramp] would visually and acoustically break up the volume" (Schön, 1987, p. 129).

In this episode, the coach feels that the student needs to explore and develop her concept more thoroughly, noting that a ramp has thickness and that this will limit her plans to use the space underneath the ramp; he urges her to draw sections. However, he does not justify this advice; as Schön puts it, he does not reveal "the meanings underlying his questions" (p. 132). Schön regards this as a hopeless confrontation in which no progress can be made on the particular design project, or on the larger project of understanding how to design. Both the student and the coach are willing to act publicly and to share actions, but they do not reflect enough on their own and one another's values and objectives, and on their interpersonal dynamics.

Reflection is not always comfortable; it forces one to consider one's own competence, to open oneself to the possibility of being wrong. Nonetheless, designers create many opportunities for reflection—for example, organizing design-review meetings, or building prototypes for formative evaluation. Such activities promote identification and integration of different perspectives; they raise concrete and detailed design issues to guide further work. In this way they help designers to reflect on the work they have already done. But they do not evoke reflection in the context of doing design. Design reviews and formative evaluations are ancillary activities that must be coordinated with design itself.

Scenarios help designers to reflect on their ideas in the context of doing design. The narrative is written to evoke an image of people doing things, pursuing goals, and using technology in support of these goals. The story enables readers to empathize with the people in the situation, which in turn leads to questions about motivations, intentions, reactions, and satisfaction. For example, in the second scenario from Table 53.1, is it valuable to Sharon to opportunistically notice other friends when she arrives at the fair? What effect does her recognition of her school gymnasium have on her success or experience at navigating the online fair?

Scenarios promote reflection and analysis in part because the human mind is adept at overloading meaning in narrative structures, both in generation and interpretation, as illustrated by the remarkable examples of dreams (Freud, 1900), myths (Levi-Strauss, 1967), and folktales (Propp, 1928/1958). It is well known that when people communicate, they rely on the given-new contract (Haviland & Clark, 1974); they assume or allude to relevant background information, then present what is novel. This normative structure eases both the generation and interpretation of narratives.

Schön (1983) described design as a "conversation" with a situation comprised of many interdependent elements. The designer makes moves and then "listens" to the design situation to understand their consequences: "In the designer's conversation with the materials of his design, he can never make a move which has only the effects intended for it. His materials are continually talking back to him, causing him to apprehend unanticipated problems and potentials" (p. 101).

When a move produces unexpected consequences, and particularly when it produces undesirable consequences, the designer articulates "the theory implicit in the move, criticizes it, restructures it, and tests the new theory by inventing a move consistent with it" (p. 155).

Scenarios often include implicit information about design consequences. Returning to the scenarios in Table 53.1, the exhibit-discussion forum in the first scenario allows Sharon to arrive at an exhibit "at any time," find out what has been said, and make a contribution. In the second scenario, it is Sharon's

chance noticing of Brian and Corey that sets up her teasing conversation with Alicia. At the same time these features have less positive consequences, for example the need to browse and read each of a discussion's comments in turn, or the requirement that she pan and zoom to get to the details about a project. These tradeoffs are important to the scenarios, but often it is enough to imply them (this is an aspect of the roughness property discussed earlier).

There are times, however, when it is useful to reflect more systematically on these relationships, to make them explicit. In another situation Sharon may find the archived discussion too long or disorganized to browse, or she may be distracted by friends in the gymnasium and never make it to Alicia's project. These alternative scenarios present a failure with respect to the starting goal. To understand, address, and track the variety of both desirable and undesirable consequences of the original annotation-design move, the designer might want to make explicit the relevant causal relationships in a scenario. Doing so provides another perspective on the envisioned situation (see Table 53.3).

Scenarios and analyses such as those shown in Table 53.3 can help designers move more deliberately toward specific consequences. For example, designers might decide to provide for subcategories within major categories in the topic listing, so that the convenience of the forum interaction is obtained, but so that it also scales well to large numbers of projects. Alternatively, the opportunistic encounters with other residents might be considered desirable enough that efforts would be made to provide "presence" information in an asynchronous web forum. As each elaboration is envisioned and proposed, it too is explored for further consequences and interdependencies.

Scenarios of use are multifarious design objects; they can describe designs at multiple levels of detail and with respect to multiple perspectives. In this way they can help designers reflect on several aspects of a problem situation simultaneously.

TABLE 53.3. Tradeoff Analysis of Features in the Scenarios from Table 53.1, Each Feature is Considered for Possible Positive and Negative Consequences for Users in the Source Scenario or Closely Related Alternatives. Each Combination of Feature with Its Consequences is Called a "Claim"

A. Online Science Fair on a Web Forum
Presenting online projects as a categorical list in a Web browser:
+ enables convenient browsing of projects, topic by topic
+ leverages existing hypermedia browsing and navigation skills
− may be difficult to scan or navigate when the list is very long or complex
B. Online Science Fair in a Collaborative Virtual Environment
Representing online projects and visitors as icons in an image of the gymnasium:
+ suggests that viewers can scan, move to, and "approach" people and projects
+ reinforces a school-centered context for thinking about the science fair
− may be perceptually complex; for example, it may include visual details that have no relevance to navigation or interaction
− may use image panning and zooming, not familiar interaction skills for some users

The scenarios in Table 53.1 provide a high-level task view, but they can be elaborated to convey the moment-to-moment thoughts and experiences of the actors in order to provide a more detailed cognitive view, or to explore actors' moment-to-moment actions to provide a more detailed functional view. They might also be elaborated in terms of the hardware or software components needed for implementing the envisioned functionality (Rosson & Carroll, 1995; Wirfs-Brock, 1995). Each of these variations in resolution and perspective is a permutation of a single underlying scenario. The permutations are integrated through their roles as complementary views of the same design object.

Using scenarios in this way makes them a more powerful design representation. They allow the designer the flexibility to develop and analyze key usage scenarios in great detail—for example, to describe core application functionality—while merely sketching less-critical scenarios. At the same time, designers are able to switch among multiple perspectives, for example directly integrating usability views with software views. Such a flexible and integrative design object can help designers manage the many interdependent consequences implied by design moves (Rosson & Carroll, 2000, 2000a).

Potential Pitfalls in Scenario-Based Design

While scenario envisioning and analysis can address many of the concerns of solution-first design, work with scenarios brings its own downsides. The concreteness and work-orientation that make scenarios effective as user-centered design representations may also introduce their own bias into the process, as designers come to identify with and optimize their ideas for particular actors or activity contexts; the very characteristics that make a story realistic and evocative may also lead designers to adopt too narrow a view. The analysis of consequences implied by a scenario (as in Table 53.3) helps to create a more balanced view of current design ideas, but at the same time may lead designers to focus on incremental changes to their ideas (e.g., to address a perceived downside) rather than to consider more innovative or transformational ideas. Also, like any design process that is centered on representations that are essentially textual, a scenario-based process (with its associated analysis and illustration) might lead to an unwelcome documentation burden, as designers are faced with maintaining and evolving a potentially large set of design scenarios at varying stages of refinement. In the next two sections we consider frameworks, design tools, and other methods that are aimed at reaping the benefits of scenario-based design while also helping to manage costs such as these.

SCENARIO-BASED USABILITY ENGINEERING

The concrete and work-oriented character of scenarios makes them an effective representation for human-centered design activities, particularly when these activities include participation by end-users or other stakeholders (Carroll et al., 2000; Chin, Rosson, & Carroll, 1997; Muller, 1992; Muller, Tudor, Wildman,

White, Root, & Dayton, 1995; Karat, 1995; Karat & Bennett, 1991; Rosson & Carroll, 2002). Scenarios can be quickly developed, shared, and revised; they are easily enriched with sketches, storyboards, or other mockups (Erickson, 1995; Kyng, 1995). A scenario of use can be directed at many concerns in system development, including documentation design and object-oriented software design (Carroll, 1995, 2000). Given these many virtues, it is no surprise that scenarios are pervasive in software design and development (Rosson, Maass, & Kellogg, 1989; Weidenhaupt, Pohl, Jarke, & Haumer, 1998). In this section we expand on the view that scenarios are a user-centered design representation. We describe a programmatic framework for employing scenarios of use in interactive system design (Carroll, 2000; for a more comprehensive presentation of the material in this section, see Rosson & Carroll, 2002).

The framework summarized in this section incorporates scenario-based analysis and design into all phases of system development, from problem analysis through usability evaluation and iterative development. We build on the general rationale for scenario-based design described earlier, but at the same time show how to make the impacts of scenario-based reasoning comprehensive and systematic. The overall process is one of usability engineering, where the scenarios support continual assessment and elaboration of the system's usefulness, ease of use, and user satisfaction. The aim is to develop a rich understanding of current activities and work practices, and to use this understanding as a basis for activity transformation.

Figure 53.1 provides an overview of the scenario-based usability engineering framework (SBUE). We assume that system development begins with an initial vision or charter, even though this may be quite sketchy or nonbinding. This vision motivates a period of intense analysis during which the current situation is examined for problems and opportunities that might be addressed by available technologies. The analysts' understanding of the current situation is communicated in problem scenarios and claims. Problem scenarios describe prototypical stakeholders engaged in meaningful activities; the claims enumerate features of the current situation that are understood to have important consequences—both positive and negative—for the scenario actors.

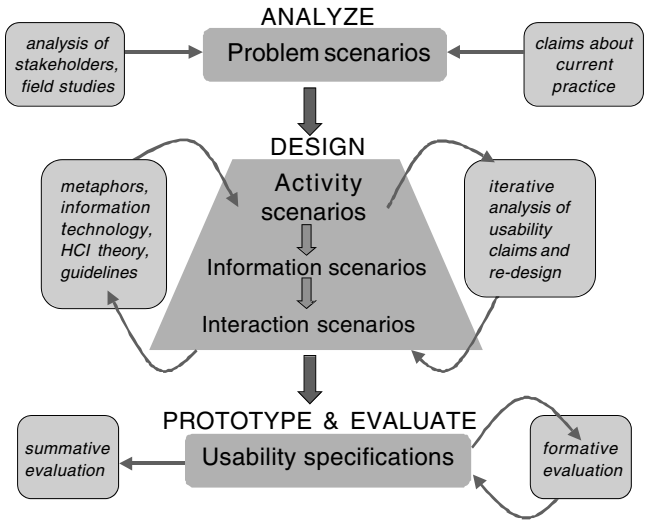


FIGURE 53.1. An overview of the scenario-based design (SBD) framework. Scenarios serve as a central representation throughout the development cycle, first describing the goals and concerns of current practices in the problem domain, and then being successively transformed and refined through an iterative design and evaluation process (from Rosson & Carroll, 2002).

The problem scenarios are transformed and elaborated through several phases of iterative design. Design envisioning is inspired by metaphors and technology options, but at the same time is constrained by the designers' knowledge of interactive system design. Each set of scenarios is complemented by claims that analyze the possible positive and negative consequences of key design features. Claims analysis leads designers to reflect on the usage implications of their design ideas while the ideas are being developed.

SBUE is mediated by evaluation activities throughout development. Each narrative serves as a test case for analytic evaluation; each claim hypothesizes usability outcomes for one or more test cases. Design scenarios are also evaluated more directly through empirical usability studies. In these the claims analysis is used for mediated evaluation, wherein the hypothesized usage impacts are operationalized and tested explicitly (Scriven, 1967). The empirical findings are interpreted with respect to the ongoing claims analysis, refining or redirecting the design efforts. We turn now to a brief example illustrating the key elements of the framework.

Problem Analysis

A challenge for any software development project is identifying the complete and correct set of requirements (Brooks, 1995). Many system requirements are functional, addressed by the actual services and information provided by the final system. Other requirements are nonfunctional, for example the measured quality of the software implementation or user interactions, or pragmatic features of the system development process like schedule, cost, or delivery platform (Rosson & Carroll, 2000b; Sommerville, 1992; Sutcliffe & Minocha; 1998). In SBUE we express an initial high-level view of requirements as a *root concept* (Table 53.4). The root concept enumerates key aspects of the team's starting vision; it is used to guide further analysis and elaboration of system requirements.

Table 53.4 contains a root concept for the Virtual Science Fair (VSF) example introduced earlier; we will use the VSF problem to illustrate the SBUE framework (see Rosson & Carroll, 2002, for a comprehensive analysis). The starting vision and rationale in this case are quite straightforward: there are obvious advantages to meeting with advisors and visitors in person, but the constraint of "same-time, same-place" limits the frequency and/or length of such meetings. Moving some of the VSF activities online increases interaction opportunity and flexibility. A side effect is that online content and communication can be stored and used for other purposes.

The root concept also documents the team's shared beliefs about the project's major stakeholders. A *stakeholder* is any person or organization who will be affected (either positively or negatively) by the system (Checkland, 1981; Muller, 1992). It

TABLE 53.4. A Root Concept for a Virtual Science Fair

Component	Contributions to the Root Concept
High-level vision	Project authors and visitors interact about science exhibits anytime, anywhere
Basic rationale	Network-based interaction overcomes barriers of place and time
	Digital media are convenient for posting, organizing, and sharing multimedia content
Stakeholder group:	
Exhibit author	Flexible tools guide construction and exhibiting of projects, and support interaction
	Ongoing access and commenting during project construction
Student advisor	Self-paced exploration of projects and interaction with others present concurrently
Fair visitor	
Starting assumptions	Open-ended participatory design process with students, advisors, and visitors
	Target users have pervasive access to personal computers and network connections
	Science fair hosted and advertised via existing community computing infrastructure

is important to take a broad view of stakeholders, particularly early in problem analysis, so that appropriate individuals and groups will be represented in the analysis activities. In the example, we consider several different sorts of people likely to be involved with VSF, so that we can consider their distinct goals with respect to system use: a student who may find the online tools helpful in planning, constructing, and talking with others about her exhibit; a student's advisor who might have more regular opportunities to provide feedback and advice; visitors to VSF who now have the option of visiting "from home," and who may benefit from other visitors' interactions with the exhibits and exhibit authors.

Although the emphasis of SBUE is on analyzing and developing usable functionality, there may be a range of nonfunctional concerns that will constrain development. These are documented as starting assumptions in the root concept. For our example project, we assume that the design of VSF will involve considerable participation by stakeholders, that the participating students, teachers, and community members have the computing resources needed to access the system, and that the fair will be integrated into a town's existing community-computing infrastructure.

The root concept lays the groundwork for an analysis of how science fairs currently take place. This might involve fieldwork—for example, a visit to a science lab where projects are being constructed. It may be more modest, perhaps a survey of participating students aimed at eliciting descriptions of activities, or a series of semi-structured interviews with students, their advisors, and their parents. One important source of information about the problem domain are the artifacts used in an activity—for instance, a flyer advertising a science fair, a poster created to display a project, a judge's notebook, or a scoring sheet used to compare exhibits. Such artifacts are an excellent source of implicit information about stakeholders' values and activities (Bodker, 1992; Norman, 1988; Rosson & Carroll, 2002).

Field studies of current practices generate rich and diverse data about needs and opportunities. To direct these data productively toward design efforts, a more abstract representation of themes and relations is required. An affinity diagram (group members posting and organizing individual observations; Beyer & Holtzblatt, 1998) is very helpful in discovering these themes. Other useful techniques include diagrams of the stakeholder relationships, hierarchical task analysis of central activities, and summaries of how collected artifacts support group activities (see Chapter 2 in Rosson & Carroll, 2002).

In SBUE the result of problem analysis is a set of problem scenarios and claims. A problem scenario is a narrative of current practice that offers a synthetic view of the actors, themes, relationships, and artifacts discovered in the fieldwork. These scenarios are not design-neutral, however. Even during early analysis activities, we assume that the team has a vision of how technology might enhance current practice. The fieldwork and related analyses will inevitably be colored by this vision (Carroll, Karat, Alpert, Van Deusen, & Rosson, 1994). If the team fails to establish a vision, or creates inconsistent or contradictory visions, this too will influence requirements analysis, but in a less positive fashion.

An effective technique for generating problem scenarios is to first describe a set of hypothetical stakeholders, individuals who will represent the different sorts of people studied during the fieldwork. It is important to create a rich but realistic view of these individuals, because they form the basis for describing and later transforming current activities and experiences. Our examples focus on the experiences of Sharon, conceived as a typical high-school junior with a large set of friends who she tries to support in their various activities, and with whom she likes to socialize whenever she gets the chance.

The scenario in Table 53.5 conveys some aspects of current science fairs; it enacts a likely visit activity and simultaneously communicates findings from fieldwork. Problem scenarios like this may be based directly on an observed episode, or they may be entirely synthetic. The goal is to introduce and contextualize the themes and relationships that will guide subsequent design work. This particular story combines our view of high-school students with issues related to real-world science fairs—for example, the fact that they are scheduled for a particular place at a particular time, the protocol for arriving, orienting oneself, being greeted and integrated into a conversation, participants' ability to adapt the discussion according to the audience, and so on.

The themes and relations implicit in a scenario can be made more explicit and open for discussion by analyzing them in claims (Table 53.6). Problem claims are analyzed by identifying features of the scenario that have notable consequences for the actors' experience. This is an instance of analytic evaluation and as such is clearly guided by the expectations and biases of the evaluator. A more systematic evaluation can be obtained by asking questions of the scenario that are guided by cognitive or motivational theories of human behavior (Carroll & Rosson, 1992; Polson, Lewis, Rieman, & Wharton, 1992). The first claim captures a key aspect of Sharon's experience at the fair—meeting in-person with her friends creates a rich communication bandwidth; and both verbal and nonverbal content can be shared, which helps Sharon know when and how to participate in the conversation.

TABLE 53.5. A Problem Scenario Describing Sharon's Visit to a Traditional Science Fair

Sharon Visits Alicia's Project at the Science Fair
Sharon is a junior at Blacksburg High School. Even though she has a biology exam tomorrow, she knows that her friends Alicia and Delia are expecting her to come to their science fair, so she decides to drive over to the high school and stay up late to study when she gets back. She remembers that Alicia has been working on her "Black Holes" poster for the past couple of weeks, and Sharon is curious to see how it came out. She also figures that some of her other friends will be there and that she can hang out with them for a while. The fair is scheduled for 7 P.M. at high-school gym. Sharon hates to be the first person at events, so she waits until 7:30 to go. The fair is already pretty crowded; she notes that most of the people milling around are parents, but she eventually spies her friends Brian and Corey across the gym talking to Alicia under the "Astronomy" banner. She can see that they are having an animated discussion—Alicia is laughing—so she heads that way. Sharon is not sure what the others have been discussing, so she just waves a hello and pauses to catch the drift of the conversation. In a few seconds, Alicia turns to say "Hi" and asks her to vote on the flashiest project. Sharon admits that she hasn't had a chance to look at any others, but that Alicia's black-and-white visuals are very dramatic. She guesses from Corey's raised eyebrows and Alicia's giggle that they must have been debating Alicia's choice of color scheme. Just then Brian's mom arrives; Sharon is impressed at how quickly Alicia drops her teasing tone and conveys a professional summary of her project's goals and activities.

Once an activity feature has been marked with a claim, the analysis can be extended through "what if" reasoning, to explore other possible positive or negative consequences. For instance, what if Sharon doesn't know Brian and Corey? What if Alicia is so occupied with the boys that she never greets or includes Sharon? Scenarios naturally evoke and support this sort of questioning; the concrete setting of the scenario invites reasoning about partially overlapping alternatives and outcomes.

TABLE 53.6. Two Claims Analyzed from the Science Fair Problem Scenario. The Feature of Interest Appears in the Shaded Area; Hypothesized Positive Consequences are Prefaced With a Plus Sign and Negative Consequences With a Minus Sign

Face-to-Face Interaction with Project Exhibitors at a Science Fair . . .
+ ensures that both nonverbal and verbal communication contribute to the interaction
+ leverages many years of experience with people's perception and communication conventions
− may introduce distracting or irrelevant personal information about partners
− inhibits parallel-communication activities (e.g., among multiple parties at once)
Using the High-School Gymnasium to Host the Science Fair . . .
+ promotes a feeling of familiarity and intimacy among students from the school
+ increases the chance that visitors will be able to envision and find their way to the fair
− requires visitors to set aside time and to travel to the site for interaction
− prevents other school events from happening in parallel (e.g., a basketball game)

An important characteristic of claims analysis is that it includes both positive and negative consequences. During problem analysis, there is a tendency to focus on the difficulties or concerns of current practice—for example, as observed in activity breakdowns or contradictions (Bødker, 1992; Kuutti, 1995; Kyng, 1995; Nardi, 1996). In contrast, designers tend to focus on the likely benefits of new features. Claims analysis imposes a balanced view of problems and opportunities. With respect to understanding users' requirements, taking a balanced view like this makes us aware of aspects of the current situation that are already working well. With respect to design envisioning, it forces us to consider side effects or other undesired impacts of changes to a situation.

Problem scenarios and claims are a central outcome of problem analysis in SBUE. Note that problem scenarios do not convey requirements in the traditional sense of the term—that is, they are not a specification of features required of a system. Instead they suggest criteria for design requirements; whatever solutions designers propose should address the positive and negative user impacts expressed in the scenarios and claims. For instance, a virtual science fair might be "required" to reinforce the advantages of in-person interaction as much as possible, but at the same time address the disadvantages of physical meetings. This is quite different from specifying that it will have a synchronous communication channel. Individual features of the solution will be identified, elaborated, evaluated, revised, or discarded in an extended iterative process.

Activity Design

Design requirements emerge and are refined throughout system development (Brooks, 1995). But at some point a team understands enough about project stakeholders and their needs that they can make specific design proposals. Indeed some projects are so overdetermined that system functions are specified in advance, and problem analysis consists simply of adapting these requirements to users' characteristics, work settings, and preferences. In SBUE, the initial step toward specifying a design solution is made by envisioning how current activities might be enhanced or even completely transformed by available technologies. We deliberately minimize attention to the concrete steps of user interaction at this point, emphasizing the basic goals and motivations of the new activities (see also essential use cases as described by Constantine and Lockwood (1999)).

SBUE is activity-centered—we analyze current practice at the level of meaningful activities, and build from this to new activities (Bertelsen & Bødker, 2003; Kuutti & Arvonen, 1992; Norman, 2005). A danger in this is that the designers will focus too much on how goals are pursued in the current situation, and on understanding and responding to people's current expectations about their tasks and about technology. Thus to encourage consideration of new options and insights, we deliberately expand the "design space" when envisioning new activities. By design space, we mean the array of possible concepts and technologies that might be relevant to the problem domain, along with some

analysis of what these options might bring to the design solution (MacLean, Young, & Moran, 1989; Moran & Carroll, 1996).

Table 53.7 exemplifies two techniques useful for exploring design alternatives. The upper part of the table shows how different conceptual metaphors evoke contrasting views of stakeholder activities. Metaphors are often used deliberately in user interface design, with the hope that users will recruit them in reasoning by analogy about what a system does or how it works (Carroll & Thomas, 1982; Carroll, Mack, & Kellogg, 1988; Madsen, 1994). Here we emphasize the role of metaphors in helping designers "think outside the box" (Erickson, 1990; VerPlank, 1988). Addressing real-world activities and concerns is crucial to effective system design, but metaphoric thinking often promotes the insights of truly creative design.

An analysis of available information technology provides a complement to the metaphoric thinking. In a sense, the technology provides another set of metaphors for thinking about the activities to be supported; but in this case the analogy is with classes of software and devices that already exist (e.g., web information systems, e-mail, or database packages). At the same time, a technology-oriented analysis such as this directs the design back to many of the pragmatic concerns of software development, by enumerating possible technologies and how they might play a role in the solution. This analysis may also be very influenced by the project's starting assumptions (see Table 53.3), if, for instance, the development organization already has developed a shared electronic whiteboard tool, or has considerable experience building web-information systems.

The exploration of metaphors and technology does not generate a new activity design. Rather it provides a set of lenses for discussion. The team might consider what it would be like if the virtual science fair was designed to be like a cocktail party versus a lecture; they can argue about the relative advantages of using a structured groupware framework versus an open-ended website. These divergent discussions form a backdrop for the convergent process of scenario writing.

TABLE 53.7. Using Metaphors and Existing Information Technology to Reason About Activities

Activity-Design Features Suggested by Metaphors for a Virtual Science Fair	
Reading at the library	Self-paced, individual access to structured information
Hearing a lecture	Large audience; prepared materials; one-way communication
Visiting a museum	Array of artifacts; small groups or individuals examine, discuss
Going to a cocktail party	Friends forming subgroups; social exchange and mingling
Activity Design Features Suggested by Information Technology for a Virtual Science Fair	
A hierarchy of web pages	Mix of text and graphics, category names, and links
An e-mail distribution list (listserv)	One-way "push" communication, possibly large audience
A shared electronic whiteboard	Informal sketches with multiple people working at once
Meeting groupware	Explicit agenda, support for floor control, meeting records

The writing of activity scenarios is a creative process influenced by many factors. One or more problem scenarios may be used as a starting point—these provide a realistic context and set of goals for exhibiting at or visiting a science fair. The associated claims guide design reasoning—the general heuristic is to maintain or even enhance the positive consequences for the actors while minimizing or removing the negatives (Carroll & Rosson, 1992). The metaphors and technology exploration provide solution ideas—how our concept of project exhibits or conversations with visitors might be expanded or transformed, and how existing technology might support these activities. Of course, the designers' knowledge of human-computer interaction and of interactive system design more broadly provide important guidance—for example, knowing the relative affordances of different computer-mediated communication channels, understanding the motivational challenges of discretionary-use software, and so on.

Two contrasting activity scenarios for Sharon's visit to VSF appear in Table 53.8. Both activity designs address the goals of the actors in the problem scenario: visiting a friend's exhibit and chatting with her and with other friends. Both respond to claims analyzed for the problem scenario—for example, attempting to maintain the familiarity and intimacy of a real-world meeting at the high school, while making participation more flexible, and enabling parallel conversations. However, they address these goals and concerns with rather different views of what constitutes an online exhibit and what network-based technology might be used to support it.

The first example was influenced by the lecture and library metaphors. New exhibit materials, and comments related to existing exhibits, are automatically distributed to the visitor; this is analogous to sitting in a room, watching a lecture, and hearing questions and answers. The online material is organized into topical categories that can be browsed in a self-paced fashion, just as Sharon might browse stacks of books in a library. These metaphors are easily supported by a combination of e-mail and web pages.

The second scenario shows an influence of the cocktail party, museum, and library metaphors. It emphasizes social exchange and informal conversation, as well as responses to an assortment of VSF artifacts in the space. Sharon is able to see which of her friends are around (predicted in advance by her IM buddy list), and is to locate them (along with other visitors) when she goes to the VSF. She can view the exhibit, scan the comment board, and catch up on the group chat history, all in parallel. Her friends are referring to specific features of Alicia's exhibit and Sharon is immediately able to interpret their meaning and jump in to the conversation.

Both scenarios have attractive consequences for Sharon and her friends. Both are possible solution approaches, so how do the designers choose? Again, many pragmatic factors contribute to this decision—the kinds of software currently in use, the team's design expertise, software development timeline and resources, organizational priorities, and so on. But assuming that both solutions are genuine possibilities, the designers must also evaluate them with respect to their usage implications. One way to do this is with participatory design sessions (Carroll et al., 2000; Chin, Rosson, & Carroll, 1997; Muller, 1992; Kyng, 1995)

TABLE 53.8. Two Alternative Activity Designs for the Online Club Meeting

A. Sharon Visits the VSF Web Forum
Sharon is a junior at Blacksburg High School. Even though she has a biology exam tomorrow, she knows that her friends Alicia and Delia have been working hard to develop their online exhibits for the science fair, and even though she has browsed their projects a few times over the past couple of weeks, the judging will be tonight. She decides to take a study to visit and see how things are going on, fully expecting that a few of her other friends will be online as well. When she logs on her IM buddy lists shows that both Alicia and Delia are online, as are their friends Brian and Corey. She uses the VSF announcement e-mail to get to the VSF website. Along with a welcome notice there are science project categories and an invitation to browse the exhibits. She knows that Alicia's project is on black holes, so finds "Astronomy" and follows the link. Another web page appears with a listing of individual projects allowing her to choose her friend's project. Alicia's exhibit is organized by the traditional pieces of a science project (problem, significance, methods, etc.). There is also a discussion forum, and when she opens it she can see comments from visitors—some have signed their names, and others are anonymous. She notices one from Brian teasing Alicia about her dramatic black-and-white color scheme. Alicia has been responding to most comments; her most recent entry just a couple of minutes ago makes Sharon think she might be watching the discussion, so she adds a (signed) comment: "Very dramatic visuals!" Alicia responds immediately and opens an IM chat.
B. Sharon Visits the VSF in a Collaborative Virtual Environment
Sharon is a junior at Blacksburg High School. Even though she has a biology exam tomorrow, she knows that her friends Alicia and Delia have been working hard to develop their online exhibits for the science fair, and even though she has browsed their projects a few times over the past couple of weeks, the judging will be tonight. She decides to take a study to visit and see how things are going on, fully expecting that a few of her other friends will be online as well. When she logs on her IM buddy lists shows that both Alicia and Delia are online, as are their friends Brian and Corey. She uses the VSF announcement e-mail to get to the VSF and arrives at a familiar spot, the middle-school gymnasium (she can imagine her favorite spot in the bleachers). All of the exhibits are arranged in different areas of the gym according to their science categories (biology, forestry, astronomy, etc.). She can see who is currently at the fair, and discovers that Brian and Corey over near the astronomy table. She knows that Alicia's project is on black holes, so she figures they are probably at her exhibit right now and moves toward the table; when she gets close enough she is able to see the exhibits on the table, including Alicia's. When she arrives next to Alicia's project she sees the online poster Alicia has been working on over the past few weeks; each section contains a traditional project component (problem, significance, methods, etc). She also sees a comment board and notices that Alicia has had quite a few conversations with visitors already tonight, including some from Brian and Corey. As she expected, Alicia is also there with Brian and Corey, engaged in a group chat. Knowing Brian, she guesses that he is teasing Alicia about her black-and-white color scheme and quickly comes to Alicia's defense.

that focus on how well the alternatives suit stakeholders' needs. As with the problem scenarios, in SBUE we evaluate design scenarios with claims analyses, considering the hypothetical positive and negative consequences of design features through "what if" discussions.

A sample claim from each scenario appears in Table 53.9. In this illustrative example, the activity scenarios were written to

TABLE 53.9. Activity Claims that Draw Out Some of the Contrasts Inhering in the Two Designs

A. Sharon Visits the VSF Web Forum
Science exhibits presented as separate pages organized by topic and submitters. . . + leverage people's familiarity with categorical hierarchies + emphasize the central and permanent recognition of individual contributions to the archive − may be tedious or complex, especially if it involves browsing extensive stored archives − may discourage people from contributing more transient and informal content to an organized archive
B. Sharon Visits the VSF in a Collaborative Virtual Environment
Science exhibits visited as objects in a space inhabited by online participants. . . + Sleverage people's familiarity with real-world conversational strategies + encourages a combination of exhibit-specific and ad hoc informal exchange − requires that conversation participants be present at the same time − may be difficult for newcomers to interrupt an ongoing conversation

be similar in many respects, but the claims capture a design contrast built into the alternative designs. The web forum offers a convenient hierarchical listing of exhibit categories and page-by-page browsing of projects, whereas the collaborative space holds simulated "exhibits" that people can approach, view, and discuss in real time. The analysis helps the designers to see the relative advantages and disadvantages of an organized asynchronous browsing-and-commenting experience and a more ad hoc synchronous exchange. Such an analysis may or may not be enough to select one alternative over the other. But at the least it begins to identify user experience concerns that provide input for continuing design discussions.

It is important to note how much progress can be made even at this very early level of envisioning activities. The narratives in Table 53.8 are quite concrete and evocative; designers or their clients can readily understand what is being proposed and begin to consider the relative pros and cons of the design ideas. Although the influence of the metaphors and information technology is obvious, few details are provided about how various sorts of the information will be presented or how participants will access and interact with this information. The scenarios are "just talk"; indeed if they are shared and discussed over an interactive medium, they can easily be extended or revised as part of a real-time design review and discussion.

User-Interaction Design

As design continues, tentative decisions are made about the design direction. For this continuing example, we will elaborate the second of the two activity scenarios in Table 53.9. We have opted for the collaborative environment design because it strongly reinforces important strengths of current real world practice (e.g., familiarity, intimacy, well-learned conversation skills; see Table 53.6), while at the same time addressing some of

the problems identified in requirements analysis (e.g., the need to travel to a specific place, the possible distraction of irrelevant personal information). We now begin to elaborate the underlying activity with information and interaction details—the user interface.

As for activity design, we first explore the design space with metaphors and technology options. The metaphors applied to information design are likely to overlap with those used in activity design, but the emphasis is shifted to what the users will see and understand about the system; the concerns of this phase are similar to those in the "gulfs of evaluation and execution" discussed by Norman (1986). For instance, the three metaphors of library, museum, and cocktail party suggest these design ideas:

- *library*: information is organized like pages in books with titles; objects are organized alphabetically or by category on shelves; there are desks and chairs for reading and working; people walk around to find, pick up, and browse objects; there may be an organized index to search for specific types of objects; visitors may be able to check out objects for more extended interaction
- *museum*: the space is broken into relatively small topic-specific rooms; objects of interest are mounted on walls or tables; there is space for groups of interested viewers; descriptive titles and text explain display objects; people may be able to use an audiotape to guide their visit
- *cocktail party*: there are various "conversation areas," perhaps including a table and chairs; visitors self-organize into groups and emit a "buzz" of conversation; new arrivals are greeted with waves or smiles

Technology options explored at this phase might include hyperlinks (icons or other controls are used to navigate or access content objects), multimedia, MOOs (object-oriented multiuser environments based on a spatial model), other graphical techniques for rendering a room (a static photograph, a panorama, a three-dimensional model), as well as techniques for representing the people and objects in the space (avatars, buddy lists, texture-mapped forms and objects).

Ideas such as these are discussed and combined with the design team's experience in user-interaction design. Information and interaction design possibilities can be "tried out" in the context of various activity design scenarios, with attention directed at claims analyzed in earlier phases as well as new features and consequences. Does the design enhance intimacy among club members? Will newcomers feel welcome? Will they be able to "catch up" and participate? Does the design support parallel activities? Design inquiry such as this—and the scenarios that provide a real-world context for the reflective process—is a hallmark of scenario-based design.

Two alternative user interaction designs appear in Table 53.10. For simplicity we focus on just the central actions at Alicia's exhibit. Both scenarios offer a view into a virtual exhibit that includes multimedia content and project-specific comments. Both assume an information model that is spatial, applying the concepts of rooms, furniture, and so on as a pervasive metaphor. Both also borrow from the other more specific metaphors described earlier—for example, there is a table that holds project exhibits, the exhibits are posters that "point" to a variety of other information, the participants form groupings and are in conversation, and so on. However, the technology supporting each scenario is quite different. In the first case the visitors are interaction within a traditional text-based MOO,

TABLE 53.10. Alternative User Interaction Designs for the Online Club Meeting Scenario

A. Sharon Visits VSF in the Community MOO
[Sharon's background and goal to visit the VSF . . .] When Sharon logs on her IM buddy lists shows that both Alicia and Delia are online, as are their friends Brian and Corey. She clicks on the VSF announcement in her e-mail to open VSF. A welcome description and snapshot of the middle-school gymnasium appears (she can imagine her favorite row and seat in the bleachers). The description introduces a number of exhibit tables, including "Astronomy" and "Biology," which is where her friends' exhibits are on display. The description also tells her who is currently visiting the fair—as expected, Alicia, Delia, Brian, and Corey are all in the space somewhere. Sharon knows that Alicia's project is on black holes, so she uses the "walk to" command to approach the Astronomy table; the VSF lists the exhibits on the table, including "Alicia on Black Holes." Sharon uses the walk command to move next to Alicia's project, where she is shown an image of a poster with names for the different sections (problem, significance, methods, etc). She is also told that there is a comment board, and when she asks to "review comments" she can see that Alicia has had quite a few conversations with visitors already tonight, including a few from Brian and Corey. As she expected, Alicia is at the exhibit with Brian and Corey, using the exhibit-specific text chat. She can see that Brian is teasing Alicia about her black-and-white color scheme and quickly comes to Alicia's defense.
B. Sharon Visits VSF in the Collaborative Environment
[Sharon's background and goal to visit the VSF. . .] When Sharon logs on, her IM buddy lists shows that both Alicia and Delia are online, as are their friends Brian and Corey. She clicks on the VSF announcement in her e-mail to open VSF. She sees a panoramic view of the middle-school gymnasium with a little rotation widget that lets or look around (she finds her favorite row and seat in the bleachers). She can also see some exhibit tables with banners over them identifying them as Astronomy and Biology and so on (she remembers that Alicia is on the "Astronomy" table and Delia in "Biology"). She can also see a bunch of little red dots indicating visitors; she wonders why some are brighter than others. She guesses that the three dots over by "Astronomy" might be Alicia and some friends; sure enough, when she mouses over the dots the names "Alicia," "Corey," and "Brian" appear. So she pans and zooms to move in closer to the table. As she zooms in, she first sees about a dozen miniature exhibits appear on the table; she can tell which one is Alicia's without even mousing over because she recognizes the dramatic black-and-white "cover sheet" Alicia has been working on for the past few weeks. She pans and zooms to get in close to the project; the cover sheet is replaced by a rectangular exhibit containing several visuals representing different parts of the exhibit, including a box where people can leave comments. But Sharon's attention is drawn instead to the group chat that Alicia is having with her friends. Each person there is represented by their VSF avatar (Brian's shows his face in a grimace—typical Brian), and their chat input shows up in bubbles as they "speak." But each bubble also has a "More . . ." link, and when Sharon clicks it she can see the chat conversation, recorded in reverse order so that she can quickly scan and get up to speed. She sees that Brian is teasing Alicia about her dramatic cover page, and quickly comes to her defense.

whereas the second presents a graphical rendition of the underlying spatial model.

Either of the designs could be used to present and interact with an online science fair, but the two proposals have different implications for how participants will experience the space. In the first case, visitors' attention is directed toward a potentially long and complex stream of descriptive text that "explains" the interspersed images. The experience is rather like reading a book or a play, with different people and objects providing the content, but much of the mental experience under the control of the reader. In contrast, the graphical view offers a concrete rendition of the space, and visitors may be attracted to one or more activities taking place in parallel, in this case a real-time chat among friends.

Some of these general implications are captured by the claims in Table 53.11. As for the earlier design claims, the usage consequences analyzed do not mandate one choice over another, but rather provoke discussion of contrasting pros and cons. For example, even though the more graphical environment may seem more familiar or inviting, we note an important negative consequence, namely that the concrete view of the gym, the tables, and so on may dampen the creativity or fantasy of visitors' experiences. This may become an issue as design continues, for example we may search for ways to suggest a hybrid approach, inviting both real world and fantasy content (Cherny, 1995).

Ultimately, user interaction design comprises all aspects of how the activities' objects and actions are rendered and executed during users' activities. In many design projects, this may include special attention to the needs of new or inexperienced users. For instance, suppose that this was Sharon's first visit to VSF. How might she know that the tables held exhibits, and that she should move in closer to "see" them? In the graphical case, the visual cues provided, along with general experience in using graphical user interfaces, might be enough to cue the behavior described in the scenario. But would she know to follow the "More . . ." link to access the chat history? It is often useful to create "user documentation" versions of design scenarios, where an actor(s) is assumed to have little or no experience using the system. In SBUE, documentation design (help texts or other guidance) is also inspired by appropriate metaphors (e.g., a coach, a policeman) and technology (e.g., online tutorials, animated demonstrations). More detailed examples of scenario-based documentation design can be found in our usability-engineering textbook (see Chapter 8 in Rosson & Carroll).

TABLE 53.11 Claims Contrasting Usage Implications of the Alternative Information designs

A. Sharon Visits the VSF in the Community MOO
Textual descriptions of people and objects present in an online space . . .
+ focus visitors' attention on a single source of information about the situation and events
+ leverages visitors' experience and enjoyment with fantasy-producing textual imagery
− may feature complex interleaving of many sorts of descriptions and communications
− may be difficult to integrate text-based descriptions into a consistent spatial model
B. Sharon Visits the VSF in a 2D Graphical Environment
Two-dimensional graphical depictions of people and objects present in an online space. . .
+ leverage club members' familiarity and habits with real-world places and objects
+ promote parallel recognition and attention to spatially distributed information and activities
+ enable the use of visual information to organize behavior (e.g., locations in room or on a table)
− may be overwhelming due to the many possibilities presented in parallel
− may be confusing or disorienting due to discontinuities in spatial navigation (relative to the real world)

In the VSF design example presented here, we considered and incorporated standard user interaction technology—the familiar WIMP paradigm of windows, icons, menus, and pointing. However, SBUE can also be used to envision and analyze the implications of more novel user-interaction paradigms and devices. For example, we might consider a role for intelligent agents as part of a new VSF scenario, and contrast this to a scenario involving community-generated FAQ (frequently asked questions) repository. Or we might explore the implications of gesture or speech recognition in lieu of (or as a complement to) keyboard and mouse input. A key advantage of exploring these ideas within a scenario context is that designers are less likely to be caught up in the new technologies for their own sake; SBUE encourages designers to try out novel ideas in usage situations that at least are believable, and that are analyzed explicitly with respect to usability consequences.

Usability Evaluation

Like most user-centered design frameworks, SBUE assumes that usability evaluation begins early and continues throughout the design and development process. Any representation of a design can be evaluated. Figure 53.1 highlighted the evaluation that takes place after detailed user-interaction scenarios have been developed. This may involve relatively formal usability testing—representative users who are recruited to carry out representative tasks on early operational prototypes. Such evaluation requires sufficient progress on a design to enable construction of a prototype, though of course user-interaction prototypes can be built with special-purpose languages or tools, or may even be low-fidelity prototypes constructed of cardboard and paper (Muller, 1991; Virzi, Sokolov, & Karis, 1996).

Starting in the very early phases of design, user feedback may be obtained in informal participatory design sessions (Chin, Rosson, & Carroll, 1997; Muller, 1992). Users can be included in discussion and envisioning of activity scenarios. The design ideas are also subjected constantly to analytic evaluation through claims analysis and other design-review activities (e.g., usability inspections or cognitive walkthrough; Nielsen, 1995; Nielsen & Mack, 1994; Polson et al., 1992). All of these activities yield formative-evaluation feedback that guides changes and expansion of the design vision.

SBUE builds on Scriven's (1967) concept of *mediated evaluation*. In mediated evaluation, empirical data are collected (what Scriven terms "pay-off" evaluation), but the materials and methods of the empirical tests are guided by prior analytic eval-

uation. The analytic evaluation may have many different components (for example, an expert-based inspection) or perhaps a cognitive model constructed for a particularly complex or critical interaction sequence (Gray, John, & Atwood, 1992; Kieras, 1997). In SBUE, the primary analysis that mediates empirical testing is claims analysis. Claims written during scenario generation and discussion document the usability issues thought to be most likely to influence users' success, failure, and satisfaction; thus the claims are used as a skeleton for constructing and administering user tests. In fact, one view of a claims set is as a series of usability hypotheses that can be assessed empirically; claims also suggest why a design feature may have a particular impact on users' experiences.

Scenarios and their associated claims can be combined to create *usability specifications* (Carroll & Rosson, 1985; Good, Spine, Whiteside, & George, 1986). A usability specification is a representative task context that has been analyzed into critical subtasks, with each subtask assigned target usability outcomes. In SBUE, the design scenarios provide a realistic task context, and the associated claims provide an analysis of the scenario that is parameterized with expected or desired usability outcomes. When the prototype has sufficient functionality that it can be tested with representative users, the specified tasks are tested and the results compared to the target outcomes.

A sample usability specification developed from the VSF scenario appears in Table 53.12. The scenario serves as a usability specification in two ways. Early (or at any point) in design, representative users may be asked to simply explore a rough prototype while pursuing the open-ended goals stated in the task context. Because the actual experience of users would vary considerably in this case, it does not make sense to establish performance outcomes; indeed, formative evaluations often include instructions to think aloud, which can slow and perhaps even interfere with task performance (Mack, Lewis, & Carroll, 1983). Nonetheless, the system is considered successful in these cases if it satisfies the goals of a user enacting the scenario; collecting general ease of use or satisfaction ratings at the end of each episode can provide a simple measure of this.

When the prototype is robust enough to measure subtask times, more detailed usability specifications guide empirical testing. In the example, a set of five simple subtasks has been analyzed from the user-interaction scenario. These tasks are directly related to claims that have been developed for key design features (only a couple of these were documented here). Performance measures are established, based either on the designers' own (expert) experiences with the prototype, or on benchmark data collected from comparable systems. Satisfaction measures are constructed to assess one or more of the specific concerns raised in the claims. For example, a negative consequence of the graphical overview was the parallel complexity of all the tables, exhibits, and attendees. The usability specification tracks this issue by requiring that users' perception of the fair "complexity" is at an acceptably low level (measured by a Likert-type rating scale with a range of 1–5). The satisfaction qualities specified for the other subtasks are similarly derived from advantages or disadvantages hypothesized by claims.

Usability specifications developed in this way have two important roles in evaluation. First, they provide concrete usability objectives that can be serve as a management tool in system development—if a product team accepts these targets, then the team's usability engineers are able to insist that redesign and improvement continue until they are met (Carroll & Rosson, 1985; Good et al., 1986).

Second, the specifications tie the results of empirical evaluation directly to the usability issues raised during design. For instance, our user interaction scenario specified that Sharon locates her friend by mousing over the red dots in the VSF overview image. A positive consequence is that awareness of other people at the VSF is enhanced; a negative consequence is that the display becomes more crowded. The time it takes to locate Alicia is specified as one measure of the feature's impact, but this performance target is complemented by users' subjective reactions to the feature. For example, users might indicate a level of agreement to a statement such as: "The red dots indicating people's location at the VSF provided feedback about current activities." Problematic results with respect to either of these usability targets raise specific issues for redesign.

As development continues, more complete prototypes or initial working systems are constructed. At this point, new usage scenarios are introduced, with new actors, goals, and activity contexts. These scenarios are also subjected to claims analysis, and the resulting claims used to develop additional usability-testing materials. This is an important step in the evaluation, because it ensures that the design has not been optimized for the

TABLE 53.12. Usability Specification Developed from VSF User Interaction Scenario

Task Context: Sharon is a busy high-school student. While studying for a biology exam, she takes a quick study break to visit the VSF advertised in the school's recent e-mail, and where she knows her friends Alicia and Delia are exhibiting their science projects. When she opens the VSF, she finds Alicia's exhibit on the "Astronomy" table; she sees that Alicia is also there with her other friends Brian and Corey. She heads over to chat with them about Alicia's exhibit.

Overall Scenario Outcome:
Average rating of at least 4.0 (of 5) on *ease of use* and *satisfaction*

	Performance Targets	Satisfaction Targets
Subtask 1: Navigate to VSF	10 seconds, 0 errors	4 on *convenience*
Subtask 2: Identify general layout of the VSF	5 seconds, 0 error	1 on *complexity*
Subtask 3: Identify location of Alicia at the VSF	20 seconds, 1 error	4 on *feedback*
Subtask 4: Navigate to Alicia's exhibit on Black Holes	30 seconds, 1 error	1 on *directness*
Subtask 5: View stored chat history	5 seconds, 0 error	4.5 on *inference*

set of design scenarios that has guided development thus far. Near the end of the development cycle (or at predefined milestones), a carefully orchestrated summative evaluation is carried out, with the goal of assessing how well the system performs with respect to its usability specifications.

Scenario-Based Iterative Design

The SBUE framework is highly iterative. Although we have emphasized the importance of grounding of new activities in thorough analysis of current practice, we assume that many new possibilities will not be realized until design ideas are made concrete and exposed to actual use. At times entirely new activities may be envisioned, when breakdowns in the technical or social environment demand a radical transformation of current practice. But even for design problems needing innovative responses, an SBUE process is valuable—the problem scenarios can help designers to predict how users may need to evolve their goals and expectations if they are to adapt to the more novel activities (Orlikowski, 1992).

More commonly, software development involves a more gradual coevolution of computing technology and people's activities: technology raises new possibilities for action, and creative responses to such possibilities lead to new requirements for support (Carroll, Kellogg, & Rosson, 1991). In these coevolutionary settings, user-interaction scenarios provide a central, activity-centered thread in iterative design, serving as intermediate design products that can be generated, shared, and revised within an overarching envisioning-evaluation-refinement cycle.

APPLICATIONS OF SCENARIO-BASED DESIGN

As exemplified by books discussing scenario-based design methods (e.g., Alexander & Maiden, 2004; Carroll, 1995, 2000), one strength of scenarios is their many potential roles in system development. Scenarios are accessible to many process stakeholders, and as such can be a tremendous aid to project communication and coordination. Scenarios can be represented at varying levels of specificity and formality, making them an attractive medium for gradual evolution and specification of a system's functions. They can also capture important aspects of the nonfunctional constraints or requirements for a project. In this section we briefly survey research and practice that is exploring different applications of scenarios to the problems of software development.

Scenarios in the System-Development Lifecycle

Our example has focused on the familiar processes of requirements analysis, design, and usability evaluation, but scenario-based methods is support many diverse goals in system development (Carroll, 1997; see Fig. 53.2). Product planners present day-in-the-life scenarios to managers as design visions (Dubberly & Mitsch, 1992); requirements engineers gather workplace scenarios through direct observation and interviews, and analyze scenarios as primary data (Antón, McCracken, & Potts, 1994; Holbrook, 1990; Hsia, Samuel, Gao, Kung, Toyoshima, & Chen, 1994; Kaindl, 1997; Kuutti, 1995; Potts, 1995).

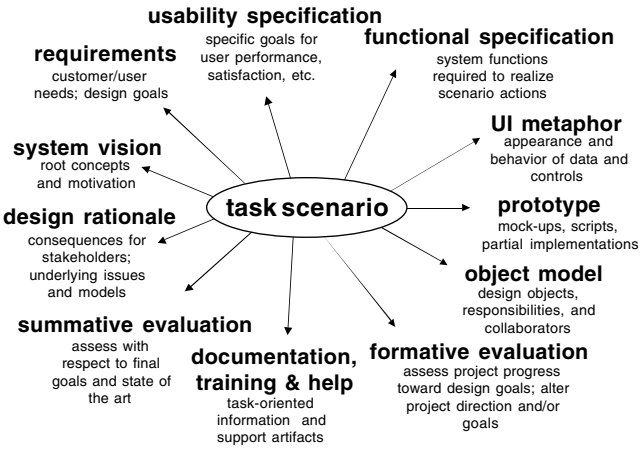


FIGURE 53.2. Scenarios have diverse uses throughout system development lifecycle.

Even if scenarios are not developed and incrementally refined as recommended in the SBUE framework, they may be used at many points along the way. For instance, task-based user documentation is often structured by scenarios. Minimalist help and training provide many examples of this, such as a "training wheels" system that blocks functions that are not relevant to a paradigmatic novice-use scenario (Carroll & Carrithers, 1983), or a "view matcher" that guides new programmers through a predefined scenario of debugging and modification (Carroll, Singer, Bellamy, & Alpert, 1990; Carroll & Rosson, 1991; Rosson & Carroll, 1996).

Personas and envisionment. The concept of a persona—the envisionment and elaboration of a hypothetical target user with her or her own personality, needs, and preferences—has been popularized by Alan Cooper and his colleagues (Cooper, 1998; Cooper & Reimann, 2003). Personas are often developed during the problem analysis or requirements specification phase of a project, as a means for understanding, expressing, and working with the goals and implied requirements of different target users. Prospective users are conveyed through detailed composite user archetypes (often represented visually, including personality characteristics), and with a context scenario that helps to anticipate user needs and expectations, so that the designers working with the personas can have as direct an experience as possible with potential users. Although it is not yet clear exactly what elements of a persona are most important in guiding or inspiring design, research does suggest that software designers find personas to be of use in the early phase of design when they are trying to understand users' needs (Vasara, 2003).

The use of personas overlaps considerably with our own SBUE framework, although we use the phrase "hypothetical stakeholder" to emphasize that these characters are imaginary, and that they should include a diverse set of perspectives and roles in the problem situation under consideration. But as for personas, SBUE encourages designers to become familiar with and empathize with envisioned characters and personal

attributes who are used repeatedly in the iterative process of scenario analysis, envisionment, and refinement.

Use cases and object-oriented design. Scenarios have also come to play a central role in object-oriented software engineering (Alexander & Maiden, 2004; Jacobson, 1995; Jacobson, Booch, & Rumbaugh, 1998; Rubin & Goldberg, 1992; Wirfs-Brock, Wilkerson, & Wiener, 1990). A use case is a scenario written from a functional point of view, enumerating all of the possible user actions and system reactions that are required to meet a proposed system function (Jacobson et al. 1992). Use cases can then be analyzed with respect to their requirements for system objects and interrelationships. Wirfs-Brock (1995) described a variant of use case analysis in which she develops a "user-system conversation"—using a two-column format, a scenario is decomposed into a linear sequence of inputs from the user and the corresponding processing and/or output generated by the system. Kaindl (2000) extended this analysis by annotating how scenario steps implement required user goals or system functions.

Scenarios are promising as a mediating representation for analyzing interactions between human-centered and software-centered object-oriented design issues (Rosson & Carroll, 1993, 1995, 2001). As we have seen, scenarios can be decomposed with respect to the software objects needed to support the narrated user interactions. These software objects can then be further analyzed with respect to their system responsibilities, identifying the information or services that should be contributed by each computational entity (Wirfs-Brock & Wilkerson, 1989; Beck & Cunningham, 1989; Rosson & Gold, 1989). This analysis (often termed "responsibility-driven design," Wirfs-Brock et al., 1990) may lead to new ideas about system functionality, for example initiatives or actions taken by a software object on behalf of the user or another object. Scenarios and claims analysis are useful in describing these new ideas and considering their usability implications in the context of use (Rosson, 1999; Rosson & Carroll, 1995; Rosson & Carroll, 2001). For instance, the VSF object might be given the responsibility to notify potential attendees when their acquaintances have arrived. A notification service such as this would help to keep attendees informed, but individuals may find the notices annoying if they appear frequently, or they may visit only when such a notice arrives.

Analyzing nonfunctional requirements. The general accessibility of scenarios makes them an excellent medium for raising and discussing a variety of competing concerns. Software engineers are concerned about issues such as code reuse, programming language or platform, and so on; management is concerned with project resources, scheduling, and so on; a marketing team focuses on issues such as the existing customer base and the product cost. These diverse concerns are nonfunctional requirements on system development—concerns about "how" a system will be developed, fielded, and maintained rather than "what" a system will provide (Sommerville, 1992). Usability goals are often specified as nonfunctional requirements, in that they typically focus on the quality of the system rather than its core functions (Mylopoulos, Chung, & Nixon, 1992). The low cost of development, content flexibility, and natural language format of scenarios and claims, make them excellent candidates for contrasting and discussing a range of such issues throughout the software-development lifecycle (Sutcliffe & Minocha, 1998). For instance, in an extension of the SBUE framework, Rosson and Carroll (2002) illustrated how a single scenario (e.g., a scenario of Sharon visiting the VSF might be re-told to include interoperability concerns related to different PC operating systems, or 24:7 accessibility concerns related to server configurations). As in SBUE, tradeoffs can be framed to raise, discuss, and prioritize competing concerns.

Taking a more formal approach, a scenario can be modeled as a sequence of task steps and this sequence can be analyzed with respect to users' ability to perform the task under different circumstances (Alexander & Maiden, 2004). For example, Gregoriades and Sutcliffe (2005) illustrated the use of Bayesian Belief Network models for assessing the reliability of human users to carry out a scenario's steps under varying environmental pressures. By noting when and where the predicted reliability fails to pass a benchmark threshold, the analysts can identify problematic aspects of a proposed scenario, as well as contrast scenarios from alternative designs. The goals of this approach overlaps with tradeoff analysis, but it provides a more systematic and mathematical representation of design issues most likely to need attention in a design.

Evaluating systems in use. In SBUE a software development process begins with fieldwork that ultimately is summarized through a set of stakeholder profiles and diagrams, themes, scenarios, and claims analysis (see Chapter 2 in Rosson & Carroll, 2002). However, researchers are currently exploring a variant of this approach in which scenarios are used as an elicitation technique for capturing current practices and reactions during the adoption and acceptance phase of a fielded system. Haynes, Purao and Skattebo (2004) reported a field study of a collaborative system for product lifecycle management in the U.S. Marine Corps. They interviewed 26 current users of the system, selecting individuals representing the various organizational roles associated with system use (e.g., engineers, scientists, staff members). Their interview script was scenario-based: after eliciting information about the interviewee's work context, they asked for personal scenarios of system use. Later they transcribed the interviews and coded the scenarios to identify system features and users' apparent concerns about these features (coded as claims with upsides and downsides). The findings were summarized and shared through focus groups to refine the evaluation. The technique allowed them to collect and summarize a number of related concerns, but also to connect (and on occasion illustrate) these concerns with the organizational roles and usage context represented by the user scenarios.

As this example suggests, it is quite possible to leverage the expressive and communicative power of scenarios in a variety of ways, even when a user-centered analyst enters a software development process that is well underway. Because scenarios are focused on the specifics of a usage context and associated user experience, they are a natural medium for "war stories" about how things are or are not working (Orr, 1986). When complemented by tradeoff analysis (e.g., documented as claims), such scenarios can become a powerful medium for discussing the need for change, whether aimed at the nature of people's goals and activities or to the technology that supports them.

Scenario-Based Design Rationale as Theory

To this point we have focused on the instrumental role of scenarios and tradeoff analyses in supporting the analysis, design, and evaluation activities of interactive software development. However, another potential role for scenario-based design is building and expressing a science base for user-centered design. In this role, a system's scenario-based analyses comprise *design rationale* (DR) that can be recruited for explanation, generalization, and reuse—contributing to the theory base for user interaction design (Sutcliffe, 2002).

Carroll and Rosson (2003) used a community MOO as a design case study that illustrates the three scientific foundations of scenario-based DR: action science, ecological science, and synthetic science. The MOO's scenarios and associated claims supported action science during the envisioning process, when novel usage features were imagined in response to the real world needs and preferences of users. The DR supported ecological science by surveying and documenting salient causal features of real world usage contexts; these surveys can guide design efforts in similar activities or domains in the future. Finally, the contributions to synthetic science came through the association of interdisciplinary evidence and explanations to the claims' upsides and downsides.

A more complete analysis of scenarios and claims as theoretical material can be found in Sutcliffe's volume on Domain Theory (Sutcliffe, 2002). He argued that claims in particular are an ideal abstraction for stating reusable knowledge about interaction design that is grounded in the relevant usage context. He continued to provide a template for developing and organizing claim-based knowledge, and argued through demonstration that such a representation is reusable from one design project to another. This potential for claims as reusable design knowledge has been explored more thoroughly by researchers developing the LINK-UP claims library and scenario-based development tool (Chewar, Bachetti, McCrickard, & Booker, 2005; Payne et al., 2003). These researchers have elaborated the claim format proposed by Sutcliffe and have been using it to collect and organize scientific knowledge about notification system design. They are currently exploring the effectiveness of the tools in teaching usability engineering concepts and skills to undergraduate and graduate students (Chewar & McCrickard, 2003).

Case-Based Learning about Scenario-Based Design

Another indirect benefit of scenario-based development is that the resulting scenarios, claims, and related design documents can support a variety of case-based learning activities. As a concrete story, even a single scenario can be evocative enough to illustrate and discuss the usage issues raised by its tradeoff analysis. A set of interrelated and successively refined scenarios and tradeoffs presents an open-ended and rich information structure for a variety of active learning experiences (Carroll & Rosson, 2005a, 2005b, 2005c).

Over the past few years, we have been collaborating with other HCI faculty and practitioners to build a case library of scenario-based usability engineering projects (ucs.ist.psu.edu), and incorporating case-based learning activities into our graduate and undergraduate teaching. The students carry out a number of in-class or homework assignments that involve case analysis over the course of a semester—for example, analyzing the implications of "perturbing" a case study in specific ways, or tracing the impacts of a tradeoff analyzed early in development through the lifecycle. They also analyze, design, and prototype their own interactive projects, documenting their work as a miniature case study. The case-based experiences serve as a sort of surrogate for apprenticeship learning in the real world, as the students are able to explore and question the reasoning and decisions of actual development teams. At the same time, the cases are an excellent vehicle for encountering and weighing the many competing concerns that development teams must address during interactive system development.

CURRENT CHALLENGES

When we design interactive systems, we design usage experiences. We create possibilities for learning, work, and leisure, for interaction and information. Scenarios—descriptions of meaningful usage episodes—are popular representational tools for making use. They help designers to understand and to create computer systems and applications as artifacts of human activity, as things to learn from, as tools to use in one's work, as media for interacting with other people.

Scenario-based design offers significant and unique leverage on some of the most characteristic and vexing challenges of design work; scenarios are at once concrete and flexible, helping developers manage the fluidity of design situations. Scenarios emphasize the context of work in the real world; this ensures that design ideas are constantly evaluated in the context of real-world activities, minimizing the risk of introducing features that satisfy other external constraints. The work-oriented character of scenarios also promotes work-oriented communication among stakeholders, helping to make design activities more accessible to many sources of expertise. Finally, scenarios are evocative, raising questions at many levels, not only about the needs of the people in a scenario as written, but also about variants illustrating design tradeoffs.

Scenario-based methods are not a panacea. A project team which complains, "We wrote scenarios, but our system still stinks!" must also report how their scenarios were developed, who reviewed them, and what roles they played in system development. If a user-interaction scenario is not grounded in what is known about human cognition, social behavior, and work practices, it may well be inspiring and evocative, but it may mislead the team into building the wrong system (Carroll et al., 1998). Scenarios are not a substitute for hard work.

At the same time, any work on user-interaction scenarios directs a project team to the needs and concerns of the people who will use a system. It is in this sense that scenarios can provide a very lightweight approach to human-centered design. Simply writing down and discussing a few key expectations about users' goals and experiences will enhance a shared vision of the problems and opportunities facing system users. Adopting a more systematic framework such as described here adds control and organization to the creative process of design, and at the

same time generates work products (scenarios and claims) that can serve as enduring design rationale during system maintenance and evolution (McKerlie & MacLean, 1994; Moran & Carroll, 1996).

Where are scenarios taking us? The current state of the art in the design of interactive systems is fragmented. Scenarios are used for particular purposes throughout system development, but there is no comprehensive process (Carroll, 1995; Jarke, Bui, & Carroll, 1998; Weidenhaupt et al., 1998). Scenario practices have emerged piecemeal, as local innovations, leading to a considerable variety of scenario types specialized for particular purposes (Campbell, 1992; Young & Barnard, 1987). A detailed textual narrative of observed workplace practices and interactions, a use-case analysis of an object-oriented domain model, a day-in-the-life video envisionment of a future product, and the instructions for test subjects in an evaluation experiment could all be scenarios. Recognizing this, and cross leveraging the many different views of scenarios, is a potential strength of scenario-based design. But much work remains in developing overarching frameworks and methods that exploit this potential advantage.

It is important for us to be ambitious, skeptical, and analytic about scenarios and scenario-based design. Forty years ago, Herman Kahn (1962) expressed puzzlement that scenarios were not more widely used in strategic planning. In the 1990s, scenarios have become so pervasive in interactive system design that younger designers may wonder what the alternative is to scenario-based design! But there is yet some strangeness to scenarios. We are not much farther than Kahn was in understanding how scenarios work as tools for planning and design, or in understanding how to fully exploit their unique strengths as aides to thought.

References

Ackoff, R. L. (1979). Resurrecting the future of operations research. *Journal of the Operations Research Society, 30*(3), 189–199.

Alexander, I. F., & Maiden, N. (2004). *Scenarios, stories, use Cases through the systems development life-cycle.* London: Wiley.

Antón, A., McCracken, W. M., & Potts, C. (1994). Goal decomposition and scenario analysis in business process reengineering. *Proceedings of CAiSE'94: Sixth Conference on Advanced Information Systems Engineering* (pp. 94–104), Springer-Verlag.

Beck, K. (1999). *Extreme programming explained: Embrace change.* Reading, MA: Addison-Wesley.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems.* San Francisco: Morgan Kaufmann.

Beck, K., & Cunningham, W. (1989). A laboratory for teaching object-oriented thinking. In N. Meyrowitz (Ed.), *Proceedings of Object-Oriented Systems, Languages and Applications: OOPSLA '89* (pp. 1–6). New York: ACM.

Bertelsen, O. W., & Bødker, S. (2003). Activity theory. In J. M. Carroll (Ed.), *HCI models, theories, and frameworks: Toward a multidisciplinary science* (pp. 291–324). San Francisco: Morgan Kaufmann.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: A customer-centered approach to system design.* San Francisco: Morgan Kaufmann.

Bødker, S. (1991). *Through the interface: A human activity approach to user interface design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Brooks, F. (1995). *The mythical man-month: Essays on software engineering* (anniversary ed.). Reading, MA: Addison-Wesley.

Campbell, R. L. (1992). Will the real scenario please stand up? *SIGCHI Bulletin, 24*(2), 6–8.

Carroll, J. M. (Ed.). (1995). *Scenario-based design: Envisioning work and technology in system development.* New York: John Wiley and Sons.

Carroll, J. M. (1997). Scenario-based design. In M. Helander & T. K. Landauer (Eds.), *Handbook of human-computer interaction* (2nd ed.; pp. 383–406). Amsterdam: North Holland.

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions.* Cambridge, MA: MIT Press.

Carroll, J. M., & Carrithers, C. (1983, November 17). *Blocking errors in a learning environment*. Paper presented at the 24th Annual Meeting of the Psychonomic Society, San Diego, CA.

Carroll, J. M., Chin, G., Rosson, M. B., & Neale, D. C. (2000). The development of cooperation: Five years of participatory design in the Virtual School. In *Proceedings of DIS 2000: Designing Interactive Systems* (pp. 239–251). New York: ACM.

Carroll, J. M., Karat, J., Alpert, S. A., van Deusen, M., & Rosson, M. B. (1994). Demonstrating raison d'etre: Multimedia design history and rationale. In C. Plaisant (Ed.), *CHI'94 Conference Companion* (pp. 29–30). New York: ACM.

Carroll, J. M., Kellogg, W. A., & Rosson, M. B. (1991). The task-artifact cycle. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 74–102). Cambridge, UK: Cambridge University Press.

Carroll, J. M., Mack, R. L., & Kellogg, W. A. (1988). Interface metaphors and user interface design. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 67–85). Amsterdam: North Holland.

Carroll, J. M., & Rosson, M. B. (1985). Usability specifications as a tool in iterative development. In H. R. Hartson (Ed.), *Advances in human-computer interaction* (pp. 1–28). Norwood, NJ: Ablex.

Carroll, J. M., & Rosson, M. B. (1990). Human-computer interaction scenarios as a design representation. In *Proceedings of the 23rd Annual Hawaii International Conference on Systems Sciences* (pp. 555–561). Los Alamitos, CA: IEEE Computer Society Press.

Carroll, J. M., & Rosson, M. B. (1991). Deliberated evolution: Stalking the view matcher in design space. *Human-Computer Interaction, 6*, 281–318.

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems, 10*, 181–212.

Carroll, J. M., & Rosson, M. B. (2003). Design rationale as theory. In J. M. Carroll (Ed.), *HCI models, theories, and frameworks: Toward an interdisciplinary science* (pp. 431–461). San Francisco: Morgan Kaufmann.

Carroll, J. M., & Rosson, M. B. (2005a). Case studies as minimalist information designs. *Proceedings of HICSS 38: Hawaii International Conference on Systems Science* (p. 406). New York: IEEE Society Press.

Carroll, J. M., & Rosson, M. B. (2005b). A case library for usability engineering: Development, experiences, and outcomes. *ACM Journal of Educational Resources in Computing, 5*(1), Article 3, 1–22.

Carroll, J. M., & Rosson, M. B. (2005c). Toward even more authentic case-based learning. *Educational Technology, 45*(6), 5–11.

Carroll, J. M., Singer, J. A., Bellamy, R. K. E., & Alpert, S. R. (1990). A view matcher for learning smalltalk. In J. C. Chew & J. Whiteside (Eds.), *Proceedings of CHI'90: Human Factors in Computing Systems* (pp. 431–437). New York: ACM.

Carroll, J. M., & Thomas, J. C. (1982). Metaphors and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man, and Cybernetics, 12*(2), 107–116.

Checkland, P. B. (1981). *Systems thinking, systems practice.* Chichester: John Wiley.

Cherny, L. (1995). Mud community. In *The mud register: Conversational modes of action in a text-based virtual reality* (pp. 42–126). Unpublished doctoral dissertation, Stanford University, Palo Alto, CA.

Chewar, C. M., Bachetti, E., McCrickard, D. S., & Booker, J. (2005). Automating a design reuse facility with critical parameters: Lessons learned in developing the LINK-UP System. In R. Jacob, Q. Limbourg, & J. Vanderdonckt (Eds.), *Computer-aided design of user interfaces IV* (pp. 235–246). Amsterdam: Kluwer Academic Publishers.

Chewar, C. M., & McCrickard, D. S. (2003). Educating novice developers of notification systems: Targeting user-goals with a conceptual framework. In *Proceedings of ED-MEDIA'03* (pp. 2759–2766). Reston, VA.

Chin, G., Rosson, M. B., & Carroll, J. M. (1997). Participatory analysis: Shared development of requirements from scenarios. In *Proceedings of Human Factors in Computing Systems, CHI'97 Conference* (pp. 162–169). New York: ACM.

Constantine, L. L., & Lockwood, L. A. D. (1999). *Software for use: A practical guide to the models and methods of usage-centered design.* Reading, MA: Addison-Wesley.

Cooper, A. (1999). *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity.* Indianapolis, IN: SAMS Press.

Cooper, A., & Reimann, R. (2003). *About Face 2.0*. Indianapolis, IN: Wiley Publishing.

Cross, N. (2001). Design cognition: Results from protocol and other empirical studies of design activity. In C. Eastman, M. McCracken, & W. Newstetter (Eds.), *Design knowing and learning: Cognition in design education* (pp. 79–103). Amsterdam: Elsevier.

Dubberly, H., & Mitsch, D. (1992). Knowledge navigator. In B. A. Myers (Ed.), *CHI'92 Special Video Program: Conference on Human Factors in Computing Systems.* New York: ACM SIGCHI.

Erickson, T. (1990). Working with interface metaphors. In B. Laurel (Ed.), *The art of human-computer interface design* (pp. 65–73). Reading, MA: Addison-Wesley.

Erickson, T. (1995). Notes on design practice: Stories and prototypes as catalysts for communication. J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 37–58). New York: John Wiley & Sons.

Freud, S. (1900). *The interpretation of dreams* (Standard ed., Vol. IV). New York: Macmillan.

Good, M., Spine, T. M., Whiteside, J., & George, P. (1986). User-derived impact analysis as a tool for usability engineering. In M. Mantei & P. Oberton (Eds.), *Proceedings of Human Factors in Computing Systems: CHI'86* (pp. 241–246). New York: ACM.

Gray, W. D., John, B. E., & Atwood, M. E. (1992). The precis of Project Ernestine, or an overview of a validation of GOMS. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), *Proceedings of Human Factors in Computing Systems: CHI'92* (pp. 307–312). New York: ACM.

Gregoriades, A., & Sutcliffe, A. (2005). Scenario-based assessment of nonfunctional requirements. *IEEE Transactions on Software Engineering, 31*(5), 392–409.

Haviland, S. E., & Clark, H. H. (1974). What's new? Acquiring new information as a process in comprehension. *Journal of Verbal Learning and Verbal Behavior, 13*, 512–521.

Haynes, S. R., Purao, S., & Skattebo, A. L. (2004). Situating evaluation in scenarios of use. In *Proceedings of CSCW 2004* (pp. 92–101). New York: ACM.

Holbrook, H. (1990, January). A scenario-based methodology for conducting requirements elicitation. *ACM SIGSOFT Software Engineering Notes, 15–1*, 95–103.

Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, Y., & Chen, C. (1994, March). Formal approach to scenario analysis. *IEEE Software, 11*(2), 33–41.

Jacobson, I. (1995). The use-case construct in object-oriented software engineering. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 309–336). New York: John Wiley & Sons.

Jacobson, I., Booch, G., & Rumbaugh, J. (1998). *The unified software development process*. Reading, MA: Addison-Wesley.

Jarke, M., Bui, X. T., & Carroll, J. M. (1998). Scenario management: An interdisciplinary approach. *Requirements Engineering, 3*(3–4), 155–173

Kahn, H. (1962). *Thinking about the unthinkable.* New York: Horizon Press.

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive Psychology, 3*, 430–454.

Kaindl, H. (1997). A practical approach to combining requirements definition and object-oriented analysis. *Annals of Software Engineering, 3*, 319–343.

Kaindl, H. (2000). A design process based on a model combining scenarios with goals and functions. *IEEE Transactions on Systems, Man, and Cybernetics, 30*(5), 537–551.

Karat, J. (1995). Scenario use in the design of a speech recognition system. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 109–133). New York: John Wiley & Sons.

Karat, J., & Bennett, J. B. (1991). Using scenarios in design meetings—a case study example. In J. Karat (Ed.), *Taking design seriously: Practical techniques for human-computer interaction design* (pp. 63–94). Boston: Academic Press.

Kieras, D. (1997). A guide to GOMS model usability evaluation using NGOMSL. In M. G. Helander, T. K. Landauer, & P. V. Pradhu (Eds.), *Handbooik of human-computer interaction* (2nd ed.; pp. 733–766). Amsterdam: North-Holland.

Kuutti, K. (1995) Work processes: Scenarios as a preliminary vocabulary. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 19–36). New York: John Wiley & Sons.

Kuutti, K., & Arvonen, T. (1992). Identifying potential CSCW applications by means of activity theory concepts: A case example. In J. Turner & R. Kraut (Eds.), *Proceedings of Computer-Supported Cooperative Work: CSCW'92* (pp. 233–240). New York: ACM.

Kyng, M. (1995). Creating contexts for design. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 85–107). New York: John Wiley & Sons.

Lévi-Strauss, C. (1967). *Structural anthropology.* Garden City, NY: Anchor Books.

Mack, R. L., Lewis, C. H., & Carroll, J. M. (1983). Learning to use office systems: Problems and prospects. *ACM Transactions on Office Information Systems, 1*, 254–271.

MacLean, A., Young, R. M., & Moran, T. P. (1989). Design rationale: The argument behind the artifact. In *Proceedings of Human Factors in Computing Systems: CHI'89* (pp. 247–252). New York: ACM.

Madsen, K. H. (1994). A guide to metaphorical design. *Communications of the ACM, 37*(12), 57–62.

McKerlie, D., & MacLean, A. (1994). Reasoning with design rationale: Practical experience with Design Space Analysis. *Design Studies, 15*, 214–226.

Medin, D. L., & Schaffer, M. M. (1978). A context theory of classification learning. *Psychological Review, 85*, 207–238.

Moran, T., & Carroll, J. M. (Eds.). (1996). *Design rationale: Concepts, techniques, and use.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Muller, M. K. (1992). Retrospective on a year of participatory design using the PICTIVE technique. In A. Janda (Ed.), *Proceedings of Human Factors of Computing Systems: CHI'92* (pp. 455–462). New York: ACM.

Muller, M. J., Tudor, L. G., Wildman, D. M., White, E. A., Root, R. A., Dayton, et al. (1995). Bifocal tools for scenarios and representations in participatory activities with users. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 135–163). New York: John Wiley & Sons.

Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering, 18*(6), 483–497.

Nardi, B. A. (Ed.). (1996). *Context and consciousness: Activity theory and human-computer interaction.* Cambridge, MA: MIT Press.

Nielsen, J. (1995). Scenarios in discount usability engineering. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 59–83). New York: John Wiley & Sons.

Nielsen, J., & Mack, R. L. (1994). *Usability inspection methods.* New York: John Wiley & Sons.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. D. Draper (Eds.), *User centered system design* (pp. 31–61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic Books.

Norman, D. A. (2005, July/August). Human-centered design considered harmful. *Interactions, 12*(4), 14–18.

Orlikowski, W. J. (1992). Learning from notes: Organizational issues in groupware implementation. In J. Turner & R. Kraut (Eds.), *CSCW'92: Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 362–369). New York: ACM.

Orr, J. (1986). Narratives at work: Story telling as cooperative diagnostic activity. In *Proceedings of CSCW'96* (pp. 62–72). New York: ACM.

Payne, C., Allgood, C. F., Chewar, C. M., Holbrook, C., & McCrickard, D. S. (2003). Generalizing interface design knowledge: Lessons learned from developing a claims library. In *Proceedings of IRI'03* (pp. 362–369). New York: IEEE.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies, 36*, 741–773.

Potts, C. (1995). Using schematic scenarios to understand user needs. In *Proceedings of ACM Symposium on Designing Interactive Systems: DIS'95* (pp. 247–256). New York: ACM Press.

Propp, V. (1958). *Morphology of the folktale*. The Hague: Mouton. (Original work published 1928)

Rosch, E., Mervis, C. B., Gray, W., Johnson, D., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology, 7*, 573–605.

Rosson, M. B. (1999, January). Integrating development of task and object models. *Communications of the ACM, 42*(1), 49–56.

Rosson, M. B., & Carroll, J. M. (1993). Extending the task-artifact framework. In R. Hartson & D. Hix (Eds.), *Advances in human-computer interaction* (Vol. 4; pp. 31–57) . New York: Ablex.

Rosson, M. B., & Carroll, J. M. (1995). Narrowing the gap between specification and implementation in object-oriented development. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 247–278). New York: John Wiley & Sons.

Rosson, M. B., & Carroll, J. M. (1996). The reuse of uses in Smalltalk programming. *ACM Transactions on Computer-Human Interaction, 3*(3), 219–253.

Rosson, M. B., & Carroll, J. M. (2000a). Nonfunctional requirements in scenario-based development. In *Proceedings of OZCHI 2000* (pp. 232–239). North Ryde, New South Wales, Australia: CSIRO Mathematical and Information Sciences.

Rosson, M. B., & Carroll, J. M. (2000b). Scenarios, objects, and points-of-view in user interface design. In M. van Harmelen (Ed.), *Object modeling and user interface design* (pp. 39–70). London: Addison-Wesley Longman.

Rosson, M. B., & Carroll, J. M. (2002). *Usability engineering: Scenario-based development of human-computer interaction.* San Francisco: Morgan Kaufmann.

Rosson, M. B., Carroll, J. M., & Rodi, C. M. (2004a). Case studies for teaching usability engineering. In *Proceedings of Special Interest Group on Computer Science Education* (pp. 36–40). New York: ACM.

Rosson, M. B., Carroll, J. M., & Rodi, C. M. (2004b). Teaching computer scientists to make sense. In I. Alexander & N. Maiden (Eds.), *Scenarios and use cases: Stories through the system lifecycle* (pp. 445–463). London: John Wiley.

Rosson, M. B., & Gold, E. (1989). Problem-solution mapping in object-oriented design. In N. Meyrowitz (Ed.), *Proceedings of OOPSLA'89: Conference on Object-Oriented Programming Systems, Languages, and Applications*. (pp. 7–10). New York: ACM.

Rosson, M. B., Maass, S., & Kellogg, W. A. (1989). The designer as user: Building requirements for design tools from design practice. *Communications of the ACM, 31*, 1288–1298.

Rubin, K., & Goldberg, A. (1992). Object behavior analysis. *Communications of the ACM, 35*(9), 48–62.

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action.* New York: Basic Books

Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco: Jossey-Bass.

Scriven, M. (1967). The methodology of evaluation. In R. Tyler, R. Gagne, & M. Scriven (Eds.), *Perspectives of curriculum evaluation* (pp. 39–83). Chicago: Rand McNally.

Sommerville, I. (1992). *Software engineering* (4th ed.). Reading, MA: Addison-Wesley.

Sutcliffe, A. (2002). *The domain theory: Patterns for knowledge and software reuse*. Mahwah, NJ: Erlbaum Associates.

Sutcliffe, A. G., & Minocha, S. (1998). Scenario-based analysis of nonfunctional requirements. *Proceedings of RESFQ 2000*, Pisa, Italy, June 8th–9th, 1998.

Tversky, A., & Kahneman, D. (1974). Judgments under uncertainty: Heuristics and biases. *Science, 185*, 1124–1131.

Vasara, K. (2003). *Introducing personas in a software development project.* Masters thesis, Helsinki University of Technology, Department of Computer Science & Engineering. Helsinki, Finland.

Verplank, W. L. (1988). Graphic challenges in designing object-oriented user interfaces. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 365–376). Amsterdam: North-Holland.

Virzi, R. A., Sokolov, J. L., & Karis, D. (1996). Usability problem identification using both low- and high-fidelity prototypes. In *Proceedings of Human Factors in Computing Systems: CHI'96* (pp. 236–243). New York: ACM.

Wasserman, A. I., & Shewmake, D. T. (1982). Rapid prototyping of interactive information systems. *ACM Software Engineering Notes, 7*(5), 171–180.

Weidenhaupt, K., Pohl, K., Jarke, M., & Haumer, P. (1998). Scenarios in system development: Current practice. *IEEE Software, 15*(2), 34–45.

Wertheimer, M. (1938). Laws of organization in perceptual forms. In W. D. Ellis (Ed.), *A sourcebook of gestalt psychology* (pp. 71–88). London: Paul, Trench, Trubner.

Wirfs-Brock, R. (1995). Designing objects and their interactions: A brief look at responsibility-driven design. In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 337–360). New York: John Wiley & Sons.

Wirfs-Brock, R., & Wilkerson, B. (1989). Object-oriented design: A responsibility-driven approach. In N. Meyrowitz (Ed.), *Object-Oriented Programming: Systems, Languages and Applications, Proceedings of OOPSLA'89* (pp. 71–76). New York: ACM.

Wirfs-Brock, R., Wilkerson, B., & Wiener, L. (1990). *Designing object-oriented software*. Englewood Cliffs, NJ: Prentice Hall.

Young, R. M., & Barnard, P. B. (1987). The use of scenarios in human-computer interaction research; Turbocharging the tortoise of cumulative science. In *Proceedings of CHI+GI'87: Conference on Human Factors in Computing Systems and Graphics Interface* (pp. 291–296). New York: ACM.

•54•

PARTICIPATORY DESIGN: THE THIRD SPACE IN HCI

Michael J. Muller
IBM *Research*

Introduction **1062**
Just Add Users and Stir? 1062
Major Bibliographic Sources for Participatory Design 1063
Conference Series 1063
Books 1063
Journals 1064
Websites 1064
Hybridity and the Third Space **1064**
Hybridity and HCI 1064
Participatory Design as the Third Space in HCI 1065
Participatory Design Contains Its Own Third Space 1065
Third Space: Negotiation, Shared Construction, and Collective Discovery in PD and HCI **1066**
Spaces and Places **1066**
Sitings 1066
Third space 1067
Claimed benefits 1067
Workshops 1067
Third space 1068
Claimed benefits 1068
Narrative Structures **1068**
Stories 1068
Third space 1069
Claimed benefits 1069
Photographs 1070
Third space 1070
Claimed benefits 1070
Dramas and Videos 1071
Third space 1071
Claimed benefits 1072
Games **1072**
Third space 1073
Claimed benefits 1073
Constructions **1073**
Language 1073
Third space 1074
Claimed benefits 1074
Making Descriptive Artifacts 1074
Third space 1074
Claimed benefits 1074
Low-Tech Prototypes 1074
Third space 1075
Claimed benefits 1075
Evolutionary Prototyping and Cooperative Prototyping 1075
Third space 1075
Claimed benefits 1075
Conclusion **1076**
References **1077**

INTRODUCTION

This chapter surveys methods, techniques, and practices in Participatory Design (PD) that can lead to hybrid experiences—that is, practices that take place neither in the workers' domain, nor in the software professionals' domain, but in an "in-between" region that shares attributes of both the workers' space and the software professionals' space. Recent work in cultural theory claims that this "in-between" region, or "third space," is a fertile environment in which participants can combine diverse knowledges with new insights and plans for action, to inform the needs of their organizations, institutions, products, and services. Important attributes of third space experiences include challenging assumptions, learning reciprocally, and creating new ideas, which emerge through negotiation and cocreation of identities, working languages, understandings, and relationships, and polyvocal (many-voiced) discussions across and through differences. The chapter focuses on participatory practices that share these attributes, including (a) site-selection of PD work; (b) workshops; (c) story collecting and story telling through text, photography, and drama; (d) games for analysis and design; and (e) the cocreation of descriptive and functional prototypes.

Just Add Users and Stir?

In a discussion of integrating women's perspectives into a male-dominated curriculum, Bunch (1987) noted that "you can't just added women and stir" (p. 140). It takes work, and new ways of thinking, and new kinds and methods of openness, to bring substantively new voices into a conversation. Similarly, to bring users' knowledges and perspectives directly into computer specification and design, it is necessary to do more than "just add users and stir." This chapter surveys methods that go beyond merely adding users—methods to create new settings and experiences that can assist computer professionals to work in partnership with diverse users in improving both computer technology and the understandings that make computer technologies successful in real use.

Participatory design (PD) is a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer-based activities (Greenbaum & Kyng, 1991; Muller & Kuhn, 1993; Schuler & Namioka, 1993). The field is extraordinarily diverse, drawing on fields such as (a) user-centered design, (b) graphic design, (c) software engineering, (d) architecture, (e) public policy, (f) psychology, (g) anthropology, (h) sociology, (i) labor studies, (j) communication studies, and (k) political science. This diversity has not lent itself to a single theory or paradigm of study or approach to practice (Slater, 1998). Researchers and practitioners are brought together—but are not necessarily brought into unity—by a pervasive concern for the knowledges, voices, and/or rights of end users, often within the context of software design and development, or of other institutional settings (e.g., workers in companies, corporations, universities, hospitals, governments). Many researchers and practitioners in PD (but not all) are motivated in part by a belief in the value of democracy to civic, educational, and commercial settings—a value that can be seen in the strengthening of disempowered groups (including workers), in the improvement of internal processes, and in the combination of diverse knowledges to make better services and products.

PD began in an explicitly political context, as part of the Scandinavian workplace democracy movement (e.g., Nygaard, 1975; Bjerknes, Ehn, & Kyng, 1987; Ehn & Kyng, 1987; Floyd, Mehl, Reisin, Schmidt, & Wolf, 1989; more recently, see Bjerknes & Bratteteig, 1995; Beck, 1996; Kyng & Matthiessen, 1997; Aarhus Conference, 2005; Winner, 1994). Early work took the form of experiments conducted by university researchers in alliances with organized labor (for historical overviews, see Ehn, 1993; Levinger, 1998).

Subsequent work focused on combining complex and distinct knowledges for realistic design problems. Fowles (2000), for example, wrote of transforming the "symmetry of ignorance" (mutual incomprehension between designers and users) into a complementary "symmetry of knowledge" through symmetries of participation and symmetries of learning. Similarly, Holmström (1995) analyzed a "gap in rationalities" among developers and users. I wrote about the need for translations among the coequal worlds of users and of software professionals, and the need to foster a polyvocal polity in which these various interested parties could coconstruct new concepts, meanings, and alliances (Muller, 1997a, 1997b). Suchman (2002) described her historical practice of PD as "working for the presence of multiple voices not only in knowledge production, but in the production of technologies as knowledges objectified in a particular way." Bødker and Buur (2002) noted the need to support the "many-voiced nature of design." These acknowledgements of the integrity and rationality of multiple voices and multiple knowledges (e.g., users and software professionals) are a crucial aspect of the argument of this chapter, concerning the creation of hybrid spaces between and among those diverse perspectives.

Recently, PD has achieved a status as a useful commercial tool in some settings (e.g., McLagan & Nel, 1995), with several major and influential consultancies forming their business identities around participatory methods,[1] and an increasing number of textbooks for design or IT governance based on participatory principles (Beyer & Holtzblatt, 1998; Bødker, Kensing, & Simonsen, 2004) This overall corporate and managerial "mainstreaming" of PD has been greeted by some with enthusiasm, and by others with dismay. Participatory work in the United States has sometimes been criticized as too friendly to management. Participator work on the Pacific Rim (e.g., Noro & Imada, 1991) appears to have grown out of the quality movement, and focuses much more on solving problems, and much less on changing workplace power relations.

A more recent trend has been the maturing of lifecycle approaches to participatory work. Early and somewhat experimental lifecycle models were offered by Mumford (1983) and Floyd (1993), anticipated in some ways by Checkland (1981).

[1]In the interest of fairness to other consultancies, I will not provide the names of commercial ventures.

Two more mature approaches have been offered by Beyer and Holtzblatt (1998) and Bødker and colleagues (2004).

This chapter primarily addresses methods, techniques, and practices in participatory design, with modest anchoring of those practices in theory. I will not repeat our recent encyclopedic survey of participatory practices (Muller, Hallewell Haslwanter, & Dayton, 1997). Rather, I will pursue a trend within those practices that has shown the most growth during the past years, and I will motivate my interest in that trend through recent advances in the theory of cultural studies. I will focus on participatory practices that fall in the hybrid realm between the two distinct work domains of (a) software professionals and (b) end users.

I should also say that my concern is for methods that have been *shown to work in real situations*—for example, those that address real problems in work life, education, home life, leisure, and so forth—in which the outcomes were of consequence, and in which the participants could freely choose whether to be involved in the work. I have, therefore, omitted many promising methods that have so far been explored only as in-laboratory university exercises, apparently as part of assigned coursework. I look forward more realistic explorations of these new methods, and I hope to include them in later revisions of this survey.

Major Bibliographic Sources for Participatory Design

Theory, practice, and experience in participatory design have been published in a series of conference proceedings and several major books.

Conference Series

Five important conference series have made major contributions to PD:

1. **Critical Computing.** Four conferences have been held, at 10-year intervals, in the Critical Computing series, most recently in 2005 (Aarhus Conference, 2005). Major papers from the conferences have appeared as two influential books (Bjerknes et al., 1987; Kyng & Matthiessen, 1997).
2. **IRIS Conference (Information systems Systems Research iIn Scandinavia).** The annual IRIS conference series often include sessions and individual contributions on participatory topics. *Proceedings* may be available through the IRIS Association, or online.[2]
3. **Participatory Design Conference.** The Participatory Design Conference has met on even-numbered years since 1990. *Proceedings* are published by Computer Professionals for Social Responsibility (CPSR).[3] Selected papers from several conferences have appeared in edited volumes or special journal issues (e.g., Kensing & Blomberg, 1998; Muller & Kuhn, 1993; Schuler & Namioka, 1993). Papers from recent conference years are available through the ACM Digital Library.[4]
4. **IFIP Conferences.** A number of conferences and workshops (sponsored by IFIP Technical Committee [TC] 9) have focused on selected topics within participatory design (e.g., Briefs, Ciborra, & Schneider [1983]; Clement, Kolm, Wagner [1994]; Docherty, Fuchs-Kittowski, Kolm, & Matthiessen [1987]; Gärtner & Wagner [1995]; and van den Besselaar, Clement, & Jaervinen [1991]).[5]
5. **Nordic Conferences on Human-Computer Interaction.** The NORDCHI conference series (sometimes also called NORDICHI) meets on even-numbered years, with a strong emphasis on participatory work within a broader Scandinavian context (Nordichi, 2006). Papers from 2002 and 2004 are available through the ACM Digital Library.

Major papers, panels, and tutorials on participatory design have also appeared in the CHI, CSCW, ECSCW, and DIS conference series, beginning as early as 1988 (*Proceedings* available through the Association for Computing Machinery,[6]), and in *Proceedings* of the Usability Professionals' Association[7] conference series, of the INTERACT conference series, and of the Human Factors and Ergonomics Society conference series. Several papers at the Co-Designing 2000 Conference[8] addressed participatory themes (Scrivener, Ball, & Woodcock, 2000).

Books

In addition to the books cited above, major collections of papers and/or chapters related to participatory design appeared in Carroll's (1995) volume on scenarios in user interaction (see also Carroll, 2000), Greenbaum and Kyng's (1991) *Design at Work*, and Wixon and Ramey's (1996) collection of papers on field-oriented methods (1996). Individual books that have been influential in the field include Bødker's (1990) application of activity theory to issues of participation, Ehn's (1988) account of work-oriented design, Suchman's (1987) discussion of situated action, and Beyer's and Holtzblatt's (1998) presentation of contextual inquiry and contextual design (see also Holtzblatt, chapter 49, this volume). A recent volume by Bødker and colleagues (2004) may broaden the impact of PD among information technology departments. Earlier influential works include a series of books on socio-technical theory and practice by Mumford (e.g., 1983; Mumford & Henshall, 1979/1983), as well as Checkland's (1981) soft systems methodology. Noro and Imada (1991) developed a hybrid ergonomic

[2]http://iris.informatik.gu.se/

[3]www.cpsr.org.

[4]http://portal.acm.org/dl.cfm

[5]http://www.ifip.or.at/. For TC 9, see http://www.ifip.or.at/bulletin/bulltcs/memtc09.htm.

[6]www.acm.org

[7]www.upassoc.org

[8]http://vide.coventry.ac.uk/codesigning/

approach, involving participation and quality programs, which has been influential around the Pacific Rim. For a historical PD bibliography, see the CPSR website

Journals

Three journals have carried the greatest number of PD papers: (a) Scandinavian Journal of Information Systems,[9] (b) Computer Supported Cooperative Work: The Journal of Collaborative Computing,[10] and (c) Human Computer Interaction.[11]

Websites

Computer Professionals for Social Responsibility maintains a set of PD resources at http://www.cpsr.org/prevsite/program/workplace/PD.html#resources.

HYBRIDITY AND THE THIRD SPACE

This chapter is concerned with participatory methods that occur in the hybrid space between software professionals and end users. Why is this hybrid space important?

Bhabha (1994) made an influential argument that the border or boundary region between two domains, or two spaces, is often a region of overlap or hybridity—for instance, a "third space" that contains an unpredictable and changing combination of attributes of each of the two bordering spaces. His area of concern was colonization, in which some native people find themselves caught in between their own traditional culture and the newly imposed culture of the colonizers (see also Dingawaney & Maier, 1994; Karttunen, 1994). Their continual negotiation and creation of their identities, as efforts of survival, creates a new hybrid or third culture (Bhabha, 1994; see also Lyotard, 1984) and even a third language (Anzaldúa, 1999; Bachmann-Medick, 1996). In such a hybrid space, enhanced knowledge exchange is possible, precisely because of those questions, challenges, reinterpretations, and renegotiations (Bachmann-Medick, 1996). These dialogues across differences—and, more importantly, *within differences*—are stronger when engaged in by groups, emphasizing not only a shift from assumptions to reflections, but also from individuals to collectives (Carrillo, 2000).

Bhabha's (1994) conception has become highly influential. Bachmann-Medick (1996) applied the concepts to translation theory. Grenfell (1998) interpreted concepts of hybridity in a study of living-at-the-border in multicultural education settings. Evanoff (2000) surveyed a number of theoretical applications of hybridity, from evolutionary biology to constructivist perspectives in sociology to democratic responses to intercultural ethical disagreements. He explored formulations from multiple disciplines, involving "third culture" in intercultural ethics, "third perspective" involving "dynamic inbetweenness" in Asian-Western exchanges, and a psychological "third area" in the development of a "multicultural personality."

TABLE 54.1. Summary of Claims Relating to Third Spaces

Overlap between two (or more) different regions or fields (inbetweenness)
Marginal to reference fields
Novel to reference fields
Not "owned" by any reference field
Partaking of selected attributes of reference fields
Potential site of conflicts between/among reference fields
Questioning and challenging of assumptions
Mutual learning
Synthesis of new ideas
Negotiation and (co-)creation of. . .
Identities
Working language
Working assumptions and dynamics
Understandings
Relationships
Collective actions
Dialogues across and within differences (disciplines)
Polyvocality
What is considered to be data?
What are the rules of evidence?
How are conclusions drawn?
Reduced emphasis on authority—increased emphasis on interpretation
Reduced emphasis on individualism—increased emphasis on collectivism
Heterogeneity as the norm

A summary of the claims relating to third spaces (or hybridity) appears in Table 54.1.

Hybridity and HCI

Within HCI, Suchman (2002) recently renewed her call for dialogue across boundaries between the partial perspectives of end users and developers. Suchman argued for boundary-crossing and mutual learning between these different standpoints, and appealed in part to recent developments in feminist epistemologies which argue that objectivity is the constructive outcome of an on-going dialogue among multiple perspectives.

The approach in this chapter begins with a similar recognition of diverse perspectives. Unlike Suchman's (2002) emphasis on the boundary *between* these perspectives, however, this chapter is concerned with creating regions of overlap where the perspectives can come into mutual knowledge and, potentially, alliance—with the creation of the hybrid spaces in which objectivity can emerge through constructive discussion, dialogue, negotiation, and mutual learning. Similarly, this chapter pursues a different solution from the located accountability recommended by Suchman, who sees each participant as located *within* a par-

[9]http://www.cs.auc.dk/~sjis/

[10]http://www.wkap.nl/journalhome.htm/

[11]http://hci-journal.com/

ticular perspective and interest, for example, "Organizations comprise multiple constituencies each with their own professional identities and views of others." By contrast, the methods in this chapter allow for the creation of new perspectives and new locations, and they acknowledge the possibility that each participant can make different choices at different moments about where to locate her or his perspective, standpoint, and thus, accountability. In keeping with the origins of PD in class struggle (e.g., Ehn & Kyng, 1987), Suchman focuses on opposing interests that meet across a designated divide. This chapter instead pursues the polyvocal polity that I proposed (Muller, 1997a) and the need identified by Bødker and Buur (2002; see also Buur & Bødker, 2000) to create a "meeting ground" for a "widen[ed] . . . circle of participants" that can "support the many voices being brought forth in order to create the new, and to find ways of supporting this multivoicedness."

There have been many calls within HCI for mutual or reciprocal learning in hybrid spaces (e.g., Bødker, Ehn, Kyng, Kammersgaard, & Sundblad, 1987; Bødker, Knudsen, Kyng, Ehn, & Madsen,1988; Druin, 1999; Druin et al., 2000; Ehn & Sjögren, 1991; Floyd, 1987; Kensing & Madsen, 1991; Lanzara, 1983; Mogensen & Trigg, 1992; Muller, 1997a; Muller, Wildman, & White, 1994; Mumford, 1983; Törpel & Poschen, 2002; Tscheligi et al., 1995). Beeson and Miskelly (2000) appealed to the notion of hybridity ("heterotopia") in describing workers who, like colonized peoples, deal "in a space which is not their own" (p. 2), taking limited and opportunistic actions to preserve "plurality, dissent, and moral space" (p. 1). Maher, Simoff, & Gabriel (2000) described the creation of virtual design spaces for sharing diverse perspectives. Merkel and colleagues (2004) described a need for "a new set of skills and competencies that go beyond technical design skills . . . to create conditions that encourage a collaborative design process and active reflection . . . for working with groups . . . that push on the traditional boundaries between users and designers" (pp. 7–8). In an early formulation, Lanzara (1983) suggested that:

> [A] large part of the design process, especially in large-scale projects and organizations involving several actors, is not dedicated to analytical work to achieve a solution but mostly to efforts at reconciling conflicting [conceptual] frames or at translating one frame into another. Much work of the designer is . . . concerned with . . . defining collectively what is the relevant problem, how to see it.

Tscheligi and colleagues (1995), in a panel on prototyping, considered that the "products" of prototyping include not only artifacts, but also understandings, communications, and relationships—a theme that was echoed in a more recent panel on modeling (Kaindl, Constantine, Karat, & Muller, 2001). Fanderclai (1995, 1996) captured a strong sense of possible new dynamics and new learnings in a hybrid online space. Finally, Thackara (2000) based part of his plenary address at CHI 2000 on the concept of the third space, providing a needed hybridity to HCI studies.

Participatory Design as the Third Space in HCI

In this chapter, I extend the HCI analyses surveyed in the preceding paragraphs, and apply Bhabha's (1994) perspective to the HCI problem of methods to bridge between two spaces—the world of the software professionals, and the world of the end users (see also Muller, 1997a, 1997b). As noted by Suchman (2002), each world has its own knowledges and practices; each world has well-defined boundaries. Movement from one world to the other is known to be difficult. We can see this difficulty manifested in our elaborate methods for requirements analysis, design, and evaluation—and in the frequent failures to achieve products and services that meet users' needs and/or are successful in the marketplace.

Traditional scientific practice in HCI has focused on instruments and interventions that can aid in transferring information between the users' world and the software world. Most of the traditional methods are relatively one-directional; for example, we analyze the requirements *from* the users, we deliver a system *to* the users, and we collect usability data *from* the users. While there are many specific practices for performing these operations, relatively few of them involve two-way discussions, and fewer still afford opportunities for the software professionals to be surprised—*to learn something that we didn't know we needed to know*.

The PD tradition has, from the outset, emphasized mutuality and reciprocity—often in a hybrid space that enabled new relationships and understandings. Bødker and colleagues (1988) made specific references to "the mutual validation of diverse perspectives." Floyd (1987) analyzed software practices into two paradigms, which she termed product-oriented (focused on the computer artifact as an end in itself) and process-oriented (focused on the human work process, with the computer artifact as means to a human goal). In her advocacy of balancing these two paradigms, Floyd noted that the process-oriented paradigm required mutual learning among users and developers (see also Segall & Snelling, 1996). Most of PD theories and practices require the combination of multiple perspectives—in part, because complex human problems require multiple disciplines (e.g., software expertise and work-domain expertise) for good solutions (e.g., Fowles, 2000; Holmström, 1995), and in part because the workplace democracy tradition reminds us that all of the interested parties (in the States, we would say "stakeholders") should have a voice in constructing solutions (e.g., Ehn & Kyng, 1987). In a related development, there are increasing calls for critical reflection in design, based on combining perspectives across disciplines, including the recent Aarhus Conference on Critical Computing (Aarhus Conference, 2005).

Participatory Design Contains Its Own Third Space

The preceding argument—that PD serves as a kind of third space to HCI—might be interesting, but is hardly worth a chapter in a handbook. I now turn to the question of hybridity in methods within the field of PD itself.

In their "tools for the toolbox" approach, Kensing and Munk-Madsen (1993) developed a taxonomy to analyze about thirty participatory methods (see also Kensing, Simonsen, & Bødker, 1996; and, in independent convergences on the same attribute, see Gjersvik & Hepsø, 1998; Luck, 2000; Reid & Reed, 2000). The first dimension of their taxonomy contrasted *abstract*

methods (suitable for a software professional's organization) with *concrete* methods (suitable for work with end users).[12] Muller and colleagues (Muller, White, & Wildman,1993; Muller et al., 1997) elaborated on this taxonomic dimension by asking *whose work domain served as the basis for the method* (in the States, we would call this a matter of "turf," as in "on whose turf did the work take place?"). At the *abstract* end of the continuum, the users have to enter the world of the software professionals in order to participate—e.g., rapid prototyping (Grønbæk, 1989) and quality improvement (Braa, 1996). At the *concrete* end of the continuum, the software professionals have to enter the world of the users in order to participate; for example, ethnography (Blomberg, Giacomi, Mosher, & Swenton-Wall, 1993; Crabtree, 1998; Orr & Crowfoot, 1992; Suchman & Trigg, 1991; see also Blomberg et al., chapter ••, this volume)), on-going tailoring during usage (Henderson & Kyng, 1991; MacLean, Carter, Lovstrand, & Moran, 1990), and end-user "design" by purchasing software for small companies (Krabbel & Wetzel, 1998; Robertson, 1996, 1998).

For the purposes of this chapter, we can now ask, "What about the practices that did not occur at the *abstract* or *concrete* end-points of the continuum? *What about the practices in between?*" These practices turn out to occur in an uncertain, ambiguous, overlapping disciplinary domain that does not "belong" to either the software professionals or the end users (e.g., these practices occur in neither the users' turf nor the software professionals' turf). The practices in between the extremes are hybrid practices, and constitute the third space of participatory design. As we explore hybrid methods that occur in this third space, we can look for HCI analogies of the attributes and advantages that were listed for Third Space studies in Table 54.1.

THIRD SPACE: NEGOTIATION, SHARED CONSTRUCTION, AND COLLECTIVE DISCOVERY IN PD AND HCI

In the remaining sections of the chapter, I will describe a diversity of participatory design techniques, methods, and practices that provide hybrid experiences or that operate in intermediate, third spaces in HCI. Because my theme is hybridity, I have organized these descriptions in terms, strategies, and moves that introduce novelty, ambiguity, and renewed awareness of possibilities, occurring at the margins of existing fields or disciplines (see Table 54.1). In several cases, a single report may fall into several categories. For example, Ehn and Sjögren (1991) conducted a workshop (see "Workshops" in this chapter) in which a storytelling method (see "Stories" in this chapter) provided a space in which people negotiated the naming and defining of workplace activities (see "Language" in this chapter). I hope that the strategies and moves of the PD practitioners and researchers will become clear, despite the multiple views onto individual reports.

SPACES AND PLACES

Sitings

One of the simplest parameters that can be manipulated to influence hybridity is the site of the work. At first, this appears to be a simple issue. As Robins (1999) said, "There are two approaches to participatory design: (1). Bring the designers to the workplace. (2). Bring the workers to the design room" (p. •••). This binary choice reflects the taxonomic distinctions that I previously reviewed; however, even within the binary choice, the selection of the site can be important. Fowles (2000), in a discussion of participatory architectural practice, provided an insight that can apply as well for HCI: "If possible[,] design workshops should be located in the locality of the participating group and in the School of Architecture. Bringing the public into the School helps to de-mystify the profession, and taking students in the community furthers their understanding of the problem and its context" (p. 65). Pedersen and Buur (2000), in their work on industrial sites, agreed (italics in the original):

> When collaborating with users *in our design environment* (e.g., a meeting space at the company), we can invite a number of users from different plants and learn from hearing them exchange work experiences . . . Being in a foreign environment (and with other users), users will tend to take a more general view of things.
>
> When collaborating with users *in their work context*, users tend to feel more at ease as they are on their home ground—we are the visitors. Tools and environment are physically present and easy to refer to. This makes for a conversation grounded in concrete and specific work experiences.
>
> The idea was born to create a type of design event with activities in both environments and with two sets of resources to support design collaboration.

In our study of telephone operators' work, we held our sessions at operator service offices and in research offices (Muller et al., 1995a). The work site meetings had the advantages of easy access to equipment on which we could demonstrate or experiment. During those meetings, we had a sense of being strongly tied to practice. The research site meetings were less tied to specific practices, and had a tendency to lead to more innovative ideas. Perhaps more subtly, the two different sites enfranchised different marginal participants. At the work site, it was easy to bring in additional work-domain experts (mostly trainers and procedures experts) who became adjunct members of the core analysis team for the duration of those meetings, *and* became resources for the core team afterwards. At the research site, it was easy to bring in more technology experts, as well as the graduate students who later performed data analysis. The research site meetings became an occasion of enfranchisement, contribution, and early commitment for these additional actors. Both core and adjunct members became authors of our report (Muller et al., 1995a).

[12]Their second dimension was of less interest for the purposes of this chapter.

Brandt and Grunnet (2000) also considered site selection in their Smart Tool and Dynabook projects, which were concerned with working conditions in the office and in the home, respectively. In the Smart Tool case, they conducted dramatic scenarios in the project designers' environment. In the Dynabook case, they asked people at home to create and enact scenarios in their own living areas.

Third space. In terms of hybridity, the selection of site can be a deliberate strategy to introduce new experiences and perspectives to one or more parties in the design process—a de-centering move that can bring people into positions of ambiguity, renegotiation of assumptions, and increased exposure to heterogeneity. Returning to Bhabha's original argument, site selection initially appears to be a matter of *moving across the boundary* between different work cultures, rather than *living within the boundary*. The use of *common design practices across sites*, however, makes those practices (and the membership of the design group) into a kind of movable third space. The practices and the group membership become stable features that persist across multiple sites. At the same time, the practices, and even the membership, grow and evolve with exposure to new sites and new understandings. In these ways, the practices become an evolutionary embodiment of the knowledge of the learnings of the group (e.g., Floyd, 1987; Muller, 1997a).

Claimed benefits. What have practitioners gained through site selection, within this deliberately hybrid-oriented work area? Several themes emerge:

- **Improved learning and understanding.** Fowles (2000) described a move from a "symmetry of ignorance" toward a "symmetry of knowledge" as diverse parties educated one another through a "symmetry of learning"—and even a kind of "transformation" through exposure to new ideas (see also Carmien et al., 2003). Brandt and Grunnet (2000), Pedersen and Buur (2000), and Muller and colleagues (1995b) also claimed that the selection of site led to the strengthening of the voices that were comfortable at each site.
- **Greater ownership.** Petersen and Buur (2000) noted that their procedures strengthened user involvement in their project. Fowles (2000) and Muller and Colleagues (1995b; see also Muller et al., 1994) make specific reference to increases in commitment and ownership of the evolving knowledge and design of the group.

Workshops

Workshops may serve as another alternative to the two "standard" sites that most of us think about. In PD, workshops are usually held to help diverse parties ("interested parties" or "stakeholders") communicate and commit to shared goals, strategies, and outcomes (e.g., analyses, designs, and evaluations, as well as workplace-change objectives). Workshops are often held at sites that are in a sense neutral—they are not part of the software professionals' workplace, and they are not part of the workers' workplace.

More importantly, workshops usually introduce novel procedures that are not part of conventional working practices. These novel procedures take people outside of their familiar knowledges and activities, and must be negotiated and collectively defined by the participants. Workshops are thus a kind of hybrid or third space, in which diverse parties communicate in a mutuality of unfamiliarity, and must create shared knowledges and even the procedures for developing those shared knowledges.

The best-known workshop format in PD is the Future Workshop (e.g., Kensing & Madsen, 1991; see also Bødker et al., 2004; McPhail, Costantino, Bruckmann, Barclay, & Clement, 1998; Mørch, Engen, & Åsand, 2004), Based in German civic planning (Jungk & Mullert, 1987), a Future Workshop proceeds through three stages: (a) *Critiquing* the present, (b) *Envisioning* the future, and (c) *Implementing*, or moving from the present to the future. These three activities involve participants in new perspectives on their work, and help to develop new concepts and new initiatives.

A number of workshops have focused on simple materials and informal diagrams, rather than on formal notations. Bødker and colleagues (2004) noted that, "The tools are simple diagrams or drawings with no special formalisms . . . because staff members participating in the workshops, as well as those to whom the results are later presented, typically have no experience with technical descriptions using [Information Technology]-originated formalisms" (p. 252).

Sanders (2000) described a family of "generative tools," activities that are selectively combined into Strategic Design Workshops, under an overall conceptual strategy that combines market research ("what people say"), ethnography ("what people do"), and participatory design ("what people make"). Activities include the construction of collages focused on thinking (e.g., "how do you expect your work to change in the future?"), mapping (e.g., laying out an envisioned work area on paper), feeling ("use pictures and words to show a health-related experience in your past"), and story telling (see "Stories" and "Making Descriptive Artifacts," following). Dandavate, Steiner, and William (2000) provided a case study of Sanders' method.

In a different setting, Buur, Binder, & Brandt (2000) developed a workshop in which workers carried a mock-up of a proposed new device (see the following section, "Making Non-Functional Artifacts") through an industrial plant, recording how it would be used. They then acted out a five-minute video scenario (see "Dramas," following), which they subsequently presented to other, similar worker teams in a workshop. Hultcrantz and Ibrahim (2002) used a similar method to concretize workshops similar to focus groups that were held with family members in their own homes. Pedell (2004) described a lower-tech storyboarding workshop format in which people created narratives using photographs, putting them in sequences and in many cases altering (typically through the addition of speech bubbles to show what people were thinking or doing). Monk and Howard (1998) used a similar method, with less emphasis on photographs, to develop a "rich picture" of a work domain.

Cameron (1998), too, faced a different setting and problem, and chose a workshop solution. This project dealt with safety issues in urban design in Baltimore and—like the METRAC program in Toronto (Nisonen, 1994)—invited community members to contribute their domain expertise as people who lived with

safety issues on an everyday basis. Cameron provided a manual, based on a professionally developed set of safety guidelines. Community members became community organizers, bringing the project topic and the proposed guidelines to their own constituencies. Two additional workshops refined the safety audit information from the constituencies, selected priority issues to fix, and adopted an action plan. Cameron observed that:

> One of the successful aspects of the Design for Safety workshop is that it provided a forum for a diverse group of people to productively discuss common problems and work through shared solutions and consensus. The workshops also showed that crime and safety were not solely the responsibility of the police, but that public works employees, traffic engineers, and especially residents must work together to envision as well as carry out the plan . . . Requiring that residents share the workshop information at community association meetings further assisted the transfer of responsibility from the workshop into the neighborhood.

Several other groups have developed repertoires of multiple workshops, from which they can select the type of workshop that is needed for a particular situation, site, or problem. Svanæs and Seland (2004) described six workshops; I list four formats that they considered successful here:

1. **Workshop 1.** Theatre, modeling clay, "design by accident," and improvisation with teenagers to explore "our mobile future"
2. **Workshop 2.** Theatre, brainstorming, and improvisation with a much more structured set of props (no modeling clay) for a different telecommunications project
3. **Workshops 4 and 5.** Theatre with audience-critique of performance (similar to Boal's Theatre of the Oppressed, previously described), sometimes using structured props as well as "designing on the spot" for new concepts, for a hospital communication project
4. **Workshop 6.** Videotaped field data as a point of common reference, before theatrical work similar to workshops 4 and 5.

Finally, Bødker, and colleagues (2004) described a repertoire of workshops. One subset of workshops was differentiated largely in terms of the artifact that was cocreated by the participants, such as freehand drawing (see also Monk & Howard, 1998), collages (see also Pedell, 2004; Sanders, 2000), affinity diagrams (see also Beyer & Holtzblatt, 1998), and timelines. Dray (1992) also used free-hand drawing technique, but in a round-robin brainstorming BrainDraw format in which *In* participants collaboratively drew *In* drawings, rotating the drawings throughout the group so that each drawing contained ideas created by each of the members of the group.

Less familiar artifacts were also used to define and differentiate workshops in the Bødker and colleagues (2004) survey. Dead Sea scrolls are textual descriptions of the history of a business process. Roll lists are brief textual descriptions of all of the interested parties related to a business activity or a technology artifact. Mapping (also called "mind mapping"—see e.g., T. Buzan & B. Buzan, 1996, for nonworkshop use of this technique) is the description of a problem area, business process, function, or other matter of interest in terms of a number of briefly stated concepts, connected by lines or arcs. A special version of mapping constructs a "communication map" among persons or roles. Finally, Prompted Reflections can be used similarly to Dray's (1992) Braindraw technique, to bring people with different design concepts into communication with one another.

Third space. The various workshop approaches have several commonalities. Each workshop brings together diverse participants to do common work, to produce common outcomes (especially Bødker et al., 2004), and to develop a plan of joint action (especially Kensing & Madsen, 1991; Bødker et al., 2004; McPhail et al., 1998; Mørch et al., 2004). They are thus opportunities that require mutual education, negotiation, creation of understanding, and development of shared commitments. Each workshop takes place in an atmosphere and (often) in a site that is not "native" to any of the participants. Thus, all of the participants are at a disadvantage of being outside of their own familiar settings, and they must work together to define their new circumstances and relationships. The combination of diverse voices leads to syntheses of perspectives and knowledges.

Claimed benefits. Advantages claimed for these experiences in hybridity include:

- **Development of new concepts** that have direct, practical value for product design (Dandavate, Steiner, & William, 2000; Kensing & Madsen, 1991; Sanders, 2000) or for community action (Cameron, 1998).
- **Engagement** of the interested parties ("stakeholders") in the process and outcome of the workshop.
- **Combinations of different people's ideas** into unified concepts.
- **Production of artifacts** that are the expected and useful "inputs" to the next stage of the development process (Bødker et al., 2004; Svanæs & Seland, 2004).

NARRATIVE STRUCTURES

Stories

Stories and story telling have played a major role in ethnographic work since before there was a field called "HCI" (for review, see Crabtree, 1998; Suchman & Trigg, 1991; see also Blomberg & Burrell, chapter 50, this volume). Stories have also had an important history in HCI (see Carroll, 1995; Erickson, 1996; Muller, 1999a). I will not attempt to review these areas. Rather, I will focus on those aspects of story collecting and story telling that involve the construction of third spaces and hybridity.

Stories in participatory work may function in at least three ways.[13] First, they may be used as triggers for conversation, analysis, or feedback (Salvador & Howells, 1998; Salvador &

[13]For a survey of story genres that may be used in participatory work, see Karasti, Baker, & Bowker (2002).

Sato, 1998, 1999). Second, they may be told by end users as part of their contribution to the knowledges required for understanding product or service opportunities and for specifying what products or services should do (Brandt & Grunnet, 2000; Lafreniére, 1996; Muller, 2001; Muller et al., 1995b; Noble & Robinson, 2000; Patton, 2000; Sanders, 2000; Tschudy et al., 1996). Third, they may be used by design teams to present their concept of what a designed service or product will do, how it will be used, and what changes will occur as a result (Druin, 1999; Druin et al., 2000; Ehn & Kyng, 1991; Ehn & Sjögren, 1986, 1991; Gruen, 2001; Muller et al., 1994; Sanders, 2000).

Beeson and Miskelly (1998, 2000) used hypermedia technologies to enable communities to tell their own stories, with the intention that "plurality, dissent, and moral space can be preserved" (Beeson & Miskelly, 2000, p. 1). They were concerned to allow multiple authors to reuse community materials selectively, telling different stories within a common context. The different accounts were organized according to themes, and laid out spatially on the image of a fictitious island for navigation by end users.

Their work entered several areas or aspects of hybridity. First, the authors of the stories (e.g., community members) were using hypermedia technology for the first time, and were thus in the role of learners, even while they were the owners of the stories, and were thus in the role of experts. Second, the authors wrote from their own perspectives, which were sometimes in strong conflict with one another. Third, the authors could make use of one another's materials, effectively moving away from single-author narratives and into a kind of collaborative collage of materials, which conveyed interlinked stories. Fourth, just as the community members were negotiating and defining their roles as learner-experts, the software professionals/researchers were negotiating and defining their roles as experts-facilitators-students. Törpel and Poschen (2002) described a related method of Narrative Transformation, emphasizing workers' roles as story creators, story analysts, and originators of new concepts that could be pursued through other methods in this chapter (e.g., low-tech prototyping).

A second line of practice and research has emphasized end users telling their stories using a system of paper-and-pencil, card-like templates. The earliest version was the Collaborative Analysis of Requirements and Design (CARD) technique of Tudor, Muller, Dayton, and Root (1993), later developed into a more general tool in Muller and colleagues (1995b) and further refined in Muller (2001). Lafreniére (1996) developed a related practice, Collaborative Users' Task Analysis (CUTA), repairing some of the deficits of CARD for his settings. Tschudy et al. (1996) developed their own highly visual version, PictureCARD, for a setting in which they had no language in common with the users whose stories they wished to understand.

The card-based practices used pieces of cardboard about the size of playing cards. Each card represented a component of the user's work or life activities, including user interface events (e.g., screen shots), social events (conversations, meetings) and cognitive, motivational, and affective events (e.g., the application of skill, the formation of goals or strategies, surprises and breakdowns, evaluations of work practices). The cards were used by diverse teams in analysis, design, and evaluation of work and technology. Because the cards were novel object to all the participants, they occasioned third-space questionings and negotiations, resulting in new shared understandings and co-constructions. Often, teams used the cards to prepare a kind of story board, narrating the flow of work and technology use and annotating or innovating cards to describe that work. The resulting posters formed narratives of the work that were demonstrated to be understandable to end users, corporate officers, and software professionals, and which led to insights and decisions of large commercial value (see Sanders, 2000, for a differently constructed example of storyboard posters to describe work).

Druin (1999; Druin et al., 2000) pursued a third line of storytelling research and practice, with children as design partners in a team that also included computer scientists, graphic designers, and psychologists (for other participatory work with children, see Sanders, 2000; Sanders & Nutter, 1994). Their purpose was to envision new technologies and practices in children's use of computers and related devices. They used both online storyboarding techniques and the construction of prototypes of spaces in which the jointly authored stories could be performed. This work kept everyone learning from everyone else—children learning about technologies and the storyboarding environment, adults learning about children's views and other adults' expertises, and everyone negotiating the meaning of new technological and narrative ideas, as well as their implementations.

So far, this section has addressed primarily the acquisition of stories; however, stories are also for telling to others. Sanders (2000) described the construction of storyboards based on users' experiences. Gruen (2000, 2001) described guidelines and practices through which a diverse team could begin with a concept, and then could craft a convincing and engaging story around it. Sanders' and Gruen's procedures led to hybrid experiences, in the sense that few software professionals or end users think in terms of story construction or rubrics for effective fictions. Irestig and Timpka (2002) described a method for sharing stories from small working groups with a larger audience of decision makers.

Third space. Story collecting and story telling generally require a kind of third space in which to occur. Beeson and Miskelly (1998, 2000) were specifically concerned to create a new space for story writing and story reading, and to maintain some of the most important aspects of third spaces in that new space—e.g., preservation and expression of new meanings, relationships, conflicts, multiple perspectives, and "heterotopia." The three card-based practices use unfamiliar media (the cards), and made those media central to the team's activities, thus requiring conscious attention to shared conceptualizing and defining of those media, as well as the creation of new media when needed. Druin and colleagues (2000) created new software environments and new devices to craft and implement stories of futuristic technologies. Finally, Gruen (2000, 2001) engaged diverse teams in new roles as story writers, guided by expert-derived guidelines, in the writing of professionally structured and professionally paced stories for organizational or commercial use.

Claimed benefits. The story-collecting and storytelling practices are diverse, and serve multiple purposes. A brief sum-

mary of the claims of their value to projects and products is as follows:

- **Articulation** and preservation of a diverse community's views (Beeson & Miskelly, 1998, 2000).
- **Practical application** to work analysis, task analysis, new technology innovation, and usability evaluation in commercially important products and services (Gruen, 2000, 2001; Lafreniére, 1996; Muller, 2001; Muller et al., 1995b; Sanders, 2000; Tudor et al., 1993; Tschudy et al., 1996).
- **Co-creation of new ideas** and children's articulation and self-advocacy (Druin, 1999; Druin et al., 2000).

Photographs

Stories can be told in many ways. One approach that has informed recent PD work is end-user photography. Patton (2000) noted that both (a) taking pictures and (b) organizing pictures into albums are, of course, familiar activities to most people in affluent countries. These activities allow end users to enter into a kind of native ethnography, documenting their own lives. In keeping with the issues raised in the preceding "Stories" section, it is important that the informants themselves (the end users) control both the camera and the selection of images (see Bolton, 1989, for a set of discussions of the uses and abuses of documentary photography). They thus become both authors and subjects of photographic accounts of their activities. This dual role leads to one kind of hybridity, in which the photographic activities partake of both the world of common social life, and the world of documenting and reporting on working conditions.

In an exploration of products for mobile knowledge workers, Dandavate and colleagues (2000) similarly asked their informants to take pictures as part of a documentation of the working lives. In their study, informants were also invited to construct collages of their working lives, selectively reusing the photographs (among other graphical items) in those collages. The collages were, in effect, one type of interpretation by the photographers of their own photographs. Similarly to Patton's work, Dandavate and colleagues asked their informants to go out of their conventional professional roles as office workers (but well within their roles as members of an affluent culture) in the activity of taking the photographs. Dandavate and colleagues then asked their informants to go even further out of role, through the construction of the collages based on their photographs and the interpretation of the collages. The activities were thus marginal, partaking of attributes of informal life and professional life, of familiar and unfamiliar activities. They concluded that the photographic work led to new learnings and understandings that had not been accessible through observational studies, as well as a stronger sense of ownership by their informants in the outcome of the study.

Noble and Robinson (2000) formed an alliance between an undergraduate design class at Massey University and a union of low-status service workers, developing photodocumentaries of service work. The photographs served as a kind of hybrid boundary object (Star & Griesemer, 1989); for the students, the photographs were composed artifacts of design, while for the union members, the photographs were common and casually produced snapshots. Discussions between union members and students were rich, conflicted, and productive, as they negotiated the status and meaning of these hybrid objects. These discussions—and the exhibits and posters that they produced (e.g., the collective actions of the students and the union members)—could not have been successful without mutual learning and construction of new understandings. Photodocumentaries were used by Kwok (2004) as a means of providing familiar, concrete artifacts to enable design collaborations. Mattelmäki and Batarbee (2002; see also Hulkko, Mattelmäki, Virtanen, & Keinonen, 2004) used photodocumentaries as one component of a set of user-composed diary techniques, with a subsequent user-created collages to serve as a rich source of discussions.[14]

Third space. End-user photography is an interesting case of hybridity and the production of third spaces. Photography is a good example of an "in-between" medium—one that is part of many people's informal lives (Dandavate et al., 2000; Noble & Robinson, 2000; Patton, 2000), but that is also an intensively studied medium of communication and argumentation (Bolton, 1989; Noble & Robinson, 2000). Photography occurs at the margin of most people's work, and yet can easily be incorporated into their work.

The resulting photographs in these projects have attributes of their dual worlds—they are partially informal and quotidian, and partially formal and documentary. Discussions around the photographs, and combination of the photographs into photonarratives (Kwok, 2004; Patton, 2000) or collages (Dandavate et al., 2000; Hulkko et al., 2004; Mattelmäki & Batarbee, 2002) can lead to mutual learning and new ideas, particularly through the inclusion of the voices of the photographers, the viewers, and especially the people depicted in the photographs (Noble & Robinson, 2000; see also discussion of Isomursu, Kuutti, & Vainamo, 2004, following).

Claimed benefits. The use of end-user photographs appears to be new and experimental, and there are few strongly supported claims of benefits. Informal claims of success and contribution include the following:

- **Richer, contextualized communication medium** between end users and designers. (In some cases, the designers were not, themselves, software professionals.)
- **Stronger engagement** of designers with end-users' worlds.
- **Enhanced sharing** of views and needs among end users, leading to stronger. articulation by them as a collective voice.

[14]It is noteworthy that, in the studies reviewed here, the informants made their own decisions about what was important, and therefore what they should photograph. For a discussion of issues in more conventional, researcher-directed photographic diary studies, see Carter and Mankoff, 2005.

Dramas and Videos

Drama provides another way to tell stories—in the form of theatre or of video. One of the important tensions with regard to drama in PD is the question of whether the drama is considered a finished piece, or a changeable work-in-progress.

Many PD drama-practitioners make reference to Boal's Theatre of the Oppressed (Boal, 1974/1992). Boal described theatrical techniques whose purpose was explicitly to help a group or a community find its voice(s) and articulate its position(s). The most influential of Boal's ideas was his Forum Theatre, in which a group of nonprofessional actors performs a skit in front of an audience of interested parties. The outcome of the skit is consistent with current events and trends—often to the dissatisfaction of the audience. The audience is then invited to become authors and directors of the drama, changing it until they approve of the outcome.

A second technique of interest involves the staging of a tableau (or a "frozen image," in Brandt & Grunnet, 2000), in which a group of nonprofessional actors positions its members as if they had been stopped in the middle of a play. Each member can tell what s/he is doing, thinking, planning, and hoping.

Forum Theatre was used informally in the UTOPIA project and other early Scandinavian research efforts (Ehn & Kyng, 1991; Ehn & Sjögren, 1991), addressing the question of new technologies in newspaper production. Changes in work patterns and work-group relations were acted out by software professionals in the end-users' workplace, using cardboard and plywood prototypes, in anticipation of new technologies. The workers served as the audience, and critiqued the envisioned work activities and working arrangements. The drama was carried out iteratively, with changes, until it was more supportive of the skilled work of the people in the affected job titles. The researchers made repeated visits with more detailed prototypes, again using the vehicle of a changeable drama, to continue the design dialogue with the workers. This work was widely credited with protecting skilled work from inappropriate automation, leading to a product that increased productivity while taking full advantage of workers' skills.

Brandt and Grunnet (2000) made a more formal use of Boal's Forum Theatre and "frozen images" in the two projects just described ("Sitings"). Working with refrigeration technicians in the Smart Tool project, they and the technicians enacted work dramas and tableaux around four fictitious workers, leading to insights about the technicians' work and the technological possibilities for enhanced support of that work. Here is Brandt and Grunnet's description of one use of Forum Theatre:

> [T]he stage was constructed of cardboard boxes which in a stylized way served as . . . the different locations in the scenario. At first the service mechanics sat as an audience and watched the play. After the first showing of the "performance" the refrigeration technicians were asked to comment and discuss the dramatized scenario critically . . .
>
> The role of the refrigeration technicians changed from being a passive audience into being directors with an expert knowledge. The users recognized the situations shown in the dramatized scenario . . . Because of the openness of the scenario there was a lot of "holes" to be filled out. For instance, one . . . technician explained that he preferred to solve the problems himself instead of calling his boss. This information meant that the Smart Tool should be able to help him solve his problems while being in his car . . . Another [technician] wanted to have personal information that his boss was not allowed . . . [to] access . . . (p. 14)

Incidents were analyzed through tableaux. The designers positioned themselves in the "frozen image" of the work situation, and then led a discussion of (a) the work activities that were captured in the stopped action, and (b) the work relations in which each particular tableau was embedded.

Muller et al. (1994) presented a related tutorial demonstration piece called Interface Theatre, with the stated goal of engaging a very large number of interested parties in a review of requirements and designs (e.g., in an auditorium). In Interface Theatre, software professionals acted out a user interface "look and feel" using a theatrical stage as the screen, with each actor playing the role of a concrete interface component (e.g., Kim the Cursor, Marty the Menubar, Dana the Dialoguebox).

Pedersen and Buur (2000; see also Buur et al., 2000), following previous work of Binder (1999), collaborated with industrial workers to make videos showing proposed new work practices and technologies. After a collaborative analysis of the work (see "Games," following), workers acted out their new ideas and took control of which action sequences were captured on video for subsequent explanation to other workers and management (see also Björgvinsson & Hillgren, 2004; Mørch et al., 2004). Isomursu and colleagues (2004) used more informal user-produced videos based on cell-phone video recordings. These included not only lay-ethnographic records of usage, but also user-originated dramas to illustrate hypothesized or desired aspects of usage. In the Situated and Participative Enactment of Scenarios method, G. Iacucci, C. Iacucci, and Kuutti (2002) described a projective series of improvisations with an innovative technology idea—the "magic thing"—in users' homes or workplaces (G. Iacucci & Kuutti, 2002; Kuutti, G. Iacucci, & C. Iacucci, 2002; see also Buur & Bødker, 2002; Bødker & Buur, 2002).

Finally, Salvador and Sato (1998, 1999) used acted-out dramas as triggers for questions in a setting similar to a focus group, and Howard, Carroll, Murphy, & Peck (2002) described the role of professional actors and directors in dramatizing attributes of proposed new products.

While all of these practices are loosely tied together through the use of drama, there are important contrasts. One important dimension of difference is the extent to which the drama is improvised in the situation, or scripted in advance. Boal's (1974/1992) techniques make a crucial use of improvisation by the user-audience, to change the action and outcome of the drama. This theme is most clearly seen in the work of Brandt and Grunnet (2000), Ehn and Sjögren (1986, 1991), and Muller and colleagues (1994). At the opposite extreme are video documentaries, which of course are difficult to change as a result of discussion and constructive insight.

Third space. Taken as a somewhat diverse participatory genre, the dramatic approaches provide many of the aspects of hybridity reviewed in the cultural studies introduction to this chapter. Drama brings a strong overlap of the world of end users and the world of software developers, showing concrete projections of ideas from one world into the other world—and, in

most uses, allowing modification of those ideas. Drama is marginal to the work domains of most software professionals and most end users, and thus moves all parties into an ambiguous area where they must negotiate meaning and collaboratively construct their understandings. Agreements, conflicts, and new ideas can emerge as their multiple voices and perspectives are articulated through this rich communication medium.

Claimed benefits. Similar to end-user photography, most of the theatrical work has the feel of experimentation. It is difficult to find clear statements of advantages or benefits of these practices (see "Conclusions," following). In general, practitioners and researchers made the following claims:

- **Building bridges** between the worlds of software professionals and users.
- **Enhancing communication** through the use of embodied (e.g., acted-out) experience and through contextualized narratives.
- **Engaging small and large audiences** through direct or actor-mediated participation in shaping the drama (influencing the usage and design of the technology).
- **Increasing designers' empathy** for users and their work.
- **Simulating use of not-yet-developed tools** and technologies ("dream tools," Brandt & Grunnet, 2000) to explore new possibilities.
- **Fuller understanding** by focus group members, leading to a more informed discussion.

GAMES

From theory to practice, the concept of games has had an important influence in participatory methods and techniques. Ehn's theoretical work emphasized the negotiation of language games in the course of bringing diverse perspectives together in participatory design (Ehn, 1988; for applications of this theory, see Ehn & Kyng, 1991; Ehn & Sjögren, 1986, 1991). In this view, part of the work of a heterogeneous group is to understand how to communicate with one another; of course, communication isn't really possible on a strict *vocabulary* basis, but requires an understanding of the *perspectives* and *disciplinary cultures* behind the words (Bachmann-Medick, 1996; Muller, 1997a, 1997b, 1999b). Thus, the work of heterogeneous teams is, in part, the "mutual validation of diverse perspectives" that Bødker and colleagues (1988) advocated.

Games have also been an important concept in designing practices, with the convergent strategies of enhanced teamwork and democratic work practices within the team.[15] We explained the concepts as follows (Muller et al., 1994):

> When properly chosen, games can serve as levelers, in at least two ways. First, games are generally outside of most workers' jobs and tasks. They are therefore less likely to appear to be "owned" by one worker, at the expense of the alienation of the non-owners. Second, . . . [PD] games . . . are likely to be novel to most or all of the participants. Design group members are more likely to learn games at the same rate, without large differences in learning due to rank, authority, or background. . . . This in turn can lead to greater sharing of ideas.
>
> In addition, games . . . can help groups of people to cohere together [and] communicate better. One of the purposes of games is enjoyment—of self and others—and this can both leaven a project and build commitment among project personnel. (pp. 62–63)

Derived from Ehn's (1988) theoretical foundation, Ehn and Sjögren (1986, 1991; see also Bødker, Grønbæk, & Kyng, 1993) adopted a "design-by-playing" approach, introducing several games into PD practice:

- **Carpentopoly**, a board game concerned with business issues in the carpentry industry.
- **Specification Game**, a scenario-based game based on a set of "situation cards," each of which described a workplace situation. Players (members of the heterogeneous analysis/design team) took turns drawing a card and leading the discussion of the work situation described on the card.
- **Layout Kit**, a game of floor-plans and equipment symbols, for a workers' view of how the shop floor should be redesigned (see also Bødker & Buur, 2002; Horgan, Joroff, Porter, & Schön, 1998; Klær & Madsen, 1995; and most recently Brandt & Messeter, 2004, reviewed below).
- **Organization Kit and Desktop Publishing Game**, a part of the UTOPIA project (Ehn & Kyng, 1991), in which cards illustrating components of work or outcomes of work were placed on posters, with annotations.

Pedersen and Buur (2000) extended the Layout Kit in new ways. Collaborating with workers at Danfoss, they jointly created a board game for laying out new technologies in an industrial plant:

> A map of the plant layout served as the game board. . . . Foam pieces in different colors and shapes worked as game pieces for the team to attach meaning to. . . . Often, in the beginning of the game, the placement of the piece was only accepted when touched by almost everybody. . . . The participants were forced to justify the placement, which fostered a fruitful dialogue about goals, intentions, benefits, and effects. People were asking each other such things as . . . "what if we change this?", "on our plant we do this, because . . .", "would you benefit from this?" (pp. 96–98)

The games became the foundation of the videos produced in collaboration with the workers (described above in "Dramas").

Buur and colleagues (2000) extended the Specification Game, making a game from the outcome of a participatory ethnographic analysis of work at an industrial plant. They first collected video observations from work activities, and developed a set of 60–70 video excerpts for further discussion. They next constructed a set of cards, one for each video excerpt, with

[15]For an example of games used to teach design experiences among students, see Iversen and Buur (2002).

a still-frame image from the video displayed on each card. Game participants then grouped these 60–70 cards into thematic clusters, organized their clusters, and analyzed the subsets of actions in each cluster (for a related nongame technique, see affinity diagramming in Beyer & Holtzblatt, 1998).

We took the concept of games in a different direction, for use in non-Scandinavian workplaces, by introducing several new games (Muller et al., 1994):

- **CARD**, a card game for laying out and/or critiquing an existing or proposed work/activity flow (see "Stories," previously).
- **PICTIVE**, a paper-and-pencil game for detailed screen design (Muller et al., 1995b).
- **Icon Design Game**, a guessing game for innovating new ideas for icons (this game assumes subsequent refinement by a graphic designer).
- **Interface Theatre**, for design reviews with very large groups of interested parties (see "Dramas," above).

Our games emphasized hands-on, highly conversational approaches to discussing both the user interface concept itself and the work processes that it was intended to support. We attempted to foster an informal and even playful tone, for the reasons sketched in the earlier quotation.

Recently, Brandt and Messeter (2004; see also Johansson, Fröst, Brandt, Binder, & Messeter, 2002) developed a strong sequence of games. Their User Game is based on the video-collage methods of Buur and colleagues (2000), combining brief video clips into person or role descriptions, which are then labeled evocatively by the participants. The second game in their sequence, the Landscape Game, places those user constructs into the work environment (as a board game). The Technology Game adds simple shapes that stand for technologies, again playing those shapes onto the work environment in the Landscape Game. Finally, the Scenario Game moves back to the real world, enacting possibilities based on new ideas from the preceding three games. The enactments may be video recording, both for documentary purposes and to generate further video material for another cycle of the four games.

Third space. Each of these ten games took all of its players outside of their familiar disciplines and familiar working practices, but strategically reduced the anxiety and uncertainty of the situation by using the social scaffolding of games. Each game required its players to work together through mutual learning to understand and define the contents of the game, and to interpret those contents to one another in terms of multiple perspectives and disciplines. The conventional authority of the software professionals was thus replaced with a shared interpretation based on contributions from multiple disciplines and perspectives.

Claimed benefits. Participatory design work with games has been claimed to lead to the following benefits:

- **Enhanced communication** through the combination of diverse perspectives.
- **Enhanced teamwork** through shared enjoyment of working in a game-like setting.
- **Greater freedom to experiment and explore new ideas** through flexible rules and redefinition of rules during the game.
- **Improved articulation** of the perspectives, knowledges, and requirements of workers.
- **New insights** leading to important new analyses and designs with documented commercial value.

CONSTRUCTIONS

Preceding sections have considered hybridity in participatory activities, such as sitings, workshops, stories, photography, dramas, and games. This section continues the survey of participatory practices that bring users and software professionals into unfamiliar and ambiguous "third space" settings. In this section, I focus on the collaborative construction of various concrete artifacts:

- **Physical reflections of a cocreated language** of analysis and design.
- **Descriptions of work** in unfamiliar media.
- **Low-tech prototypes** for analysis and design.
- **High-tech prototypes** for design and evaluation.

Language

The preceding section noted Ehn's (1988) theoretical work on *PD as language games*. Ehn's interest converges with Bhabha (1994) "third space" argument: Part of the characterization of hybridity was the negotiation and cocreation of working language and meaning. This section takes Ehn's position seriously, and considers the role of language creation in participatory practices that lead to hybridity.

Several projects have made physical objects into a kind of vocabulary for work analysis, design, or evaluation. The cards described in the preceding section ("Games") are examples (Buur et al., 2000; Ehn & Sjögren, 1986, 1991; Lafreniére, 1996; Muller, 2001; Muller et al., 1995b; Tschudy et al., 1994). In each of these methods, the cards became a kind of "common language" (e.g., Muller et al., 1995b) through which the design team communicated (a) with one another, and (b) with their labor and management clients.

In two of the methods, the cards themselves were acknowledged to be incomplete, and part of the work of the team was to develop and refine the cards so as to reflect their growing understanding and their new insights (Lafreniére, 1996; Muller, 2001). Team members (users and others) were encouraged to disregard, if appropriate, the template of information on each card, up to and including the decision to turn the card over and write on its blank back. In subsequent sessions, the concepts that were written on the blank backs of cards usually became new kinds of cards. The working vocabulary of the team thus grew as the shared understanding of the team grew. This extensibility of the set of cards was observed in nearly all sessions, but was particularly important in sessions that were envisioning

future technologies or future work practices. The cards thus became a point of hybridity, where assumptions were questioned and challenged, where extensive and polyvocal dialogue was required for the team to assign meaning to the cards, where conflicts were revealed and resolved, and where the team had to construct its understanding and its language.

Similarly, the board games of Ehn and Sjögren, and especially of Pedersen and Buur (2000), used deliberately ambiguous playing pieces. The analysis team had to assign meaning to the pieces, and did so in a collaborative way.

Chin, Schuchardt, Myers, and Gracio (2000), working with a community of physical scientists who were not software professionals, introduced software-like flowcharts to their clients (see Kensing & Munk-Madsen, 1993, for a discussion of the relationship between concrete tools and abstract tools). This work shared, with the other work reviewed in this section, aspects of symbol-ambiguity and language cocreation:

> To attune scientists to the construction of workflow diagrams, we provided them a simple, informal example of how a meteorologist might diagram his [sic] work in collecting and reporting weather conditions. . . . Although we used circles and arrows in our example, we did not impose any specific symbology or rules on the scientists' construction of workflow diagrams. . . . At times, the scientists did struggle in developing some diagrams, but the labor was mostly centered on the elucidation of the research processes rather than the mechanics of diagramming. (p. 32)

Third space. Common to all of these projects was the cocreation of a physically represented language, both within the team and from the team to its clients and stakeholders. This kind of lay linguistic work requires mutual education and mutual validation for the new language components to have meaning to all of the parties. These negotiations of multiple knowledges are at the heart of the "third space" proposal of Bhabha (1994).

Claimed benefits. Most of these projects involved a number of activities, and a number of aspects of hybridity. It is difficult to determine how much of their successes were due specifically to the language-related components. Benefits that *may* have resulted from the negotiation and cocreation of language include the following:

- **Enhanced understandings** of one another's perspectives and needs.
- **Critical examinations of assumptions** underlying the ways that each party expressed its perspective.
- **Shared ownership of the language** and its physical manifestation (cards, flowcharts, game pieces).
- **Improved communication** within the team and from the team to interested outsiders (clients, stakeholders).

Making Descriptive Artifacts

Another way of moving end users into unfamiliar and hence reflective experiences is to ask them to use "projective" or artistic methods to report on their experiences and needs. In one sense, these methods produce another kind of language of expression, and therefore might have been included in the preceding section. Because the outcomes are so distinctively different from the language-oriented work of the preceding section, I thought it best to review this work in its own section.

Sanders has employed user-created collage in her participatory practice for a number of years (Sanders, 2000; see also Dandavate et al., 2000; Sanders & Branaghan, 1998; Sanders & Nutter, 1994). The choice of collage is, of course, strategic; relatively few people make collages as part of their work activities, and relatively few people interpret their collages to one another as part of their work conversations. Yet the content of the collages is strongly anchored in what people know. The collages thus become marginal constructions, not part of any defined workplace field or discipline, but informed by familiar knowledges. The novelty of the collage encourages the challenging of assumptions, and the interpretation and presentation of collages encourages mutual learning across the diversity of experiences and knowledges of the participants.

For completeness, I make reference to the work of Noble and Robinson (2000) on collaborative creation of photo documentaries, and of Patton (2000) on end-user creation of photo collages, reviewed in the earlier section on "Photographs." Their work also produced descriptive artifacts that took users and their collaborators into unfamiliar areas.

Third space. These methods have in common the use of a nonstandard medium for making users' needs known, and for developing new insights in a workplace setting. The making of collages may be new for many participants. They are thus in a kind of "third space," between their work culture and the artistic or expressive culture of collages, and they have to reflect on the differences as they construct their approach to making collages of their own experiences.

It is not clear, in Sanders' work, whether the collage work is done collaboratively among end users, or whether each collage is a solitary production. If the collage-creation is done collaboratively, then it might give rise to some of the other attributes of hybridity in Table 54.1—e.g., challenging assumptions, co-creation of meanings and collective actions, dialogues.

Claimed benefits. Basing her claims on years of practice with collages and related practices, Sanders (2000) claimed the following benefits:

- **Using visual ways** of sensing, knowing, remembering, and expressing.
- **Giving access and expression to emotional side** of experience.
- **Acknowledging the subjective perspective** in people's experiences with technologies.
- **Revealing unique personal histories** that contribute to the ways that people shape and respond to technologies.

Low-Tech Prototypes

Beaudouin-Lafon and Mackay (chapter 52, this volume) have provided a chapter on prototyping—including participatory

prototyping—in this volume; therefore, I have written a very brief account in this chapter so as not to duplicate their efforts.

Low-tech prototypes may lead to "third space" experiences because they bring people into new relationships with technologies—relationships that are "new" in at least two important ways. First, the end users are often being asked to think about technologies or applications that they have not previously experienced. Second, in *participatory* work with low-tech prototypes, end users are being asked to use the low-tech materials to reshape the technologies—a "design-by-doing" approach (Bødker et al., 1993). In this way, participatory work with low-tech prototypes involves much more user contribution and user initiative than the more conventional use of "paper prototypes" as surrogates for working systems in usability testing (e.g., Daly-Jones, Bevan, & Thomas, 1999; Rettig, 1994).

The UTOPIA project provided impressive demonstrations of the power of low-tech cardboard and plywood prototypes to help a diverse group to think about new technologies, office layouts, and new working relations that might result from them (Bødker et al., 1987, 1988, 1993; Ehn & Kyng, 1991; for other use of low-tech, substitutive prototypes, see Mørch et al., 2004). Subsequent projects to translate this work to North America led to the PICTIVE method of paper-and-pencil constructions of user interface designs by heterogeneous design teams (Muller et al., 1995b); prototyping of consumer appliances using foam-core and hook-and-loop attachments (Sanders & Nutter, 1994); and a more experimental simulation of email, using paper airplanes (Dykstra & Carasik, 1991).

Third space. Low-tech prototyping has a reputation for bringing new insights through the combination of diverse perspectives. The UTOPIA project is widely credited with mutual education among shop-floor print workers and computer systems researchers. Our experiences with PICTIVE almost always involved mutual education. Understanding and changing the artifact become important arenas for people to explore their understandings of one another's positions, to question one another's approaches, to discover and resolve conflicts, to engage in combinations of views leading to plans for collective action, and to accommodate heterogeneity of views and interests.

Claimed benefits. The low-tech participatory prototyping approaches have been extraordinarily influential, with adoption on four continents. Claimed benefits include

- **Enhanced communication and understanding** through grounding discussions in concrete artifacts.
- **Enhanced incorporation of new and emergent ideas** through the ability of participants to express their ideas directly via the low-tech materials, and through the construction of artifacts that can be used in other techniques, especially drama and video documentaries.
- **Enhanced working relations** through a sense of shared ownership of the resulting design.
- **Practical application with measured successes** in using low-tech design approaches to real product challenges, achieving consequential business goals.

Evolutionary Prototyping and Cooperative Prototyping

This last section on participatory methods is concerned with software prototyping. As noted above, I am relying on Beaudouin-Lafon and Mackay's chapter 15 in this volume to cover prototyping in greater depth and breadth. I include this brief overview for completeness of my chapter's survey of hybridity in participatory practices.

Bødker and Grønbæk (1991) and Madsen and Aiken (1993) explored the potential of cooperative prototyping in several projects, using different technology infrastructures. In general, they found that this approach led to enhanced communication with end users, improved incorporation of end-user insights into the prototypes, and stronger collective ownership and collective action planning by the team. They also observed time-consuming breakdowns in the design process itself, when new ideas required significant programming effort.

In a different prototyping approach, a system is delivered to its end users as series of iterative prototypes, each of which gradually adds functionality (e.g., Anderson & Crocca, 1993; Bertelsen, 1996; Trigg, 2000). What appears to be critical is that the prototype functions as a *crucial artifact* in the end-users' work, such as, (a) a resource of documents for librarians (Anderson & Crocca, 1993), (b) an online event checklist that served as the crucial coordination point for the work of diverse contributions (Bertelson, 1996), or (c) a database supporting funding work in a nonprofit organization (Trigg, 2000). Trigg (2000) provided a series of observations and tactical recommendations about how to engage the users in the evaluations that both they and the software professionals had agreed were needed.

Third space. This very brief survey of cooperative prototyping and "iterative delivery" approaches shows several aspects of hybridity. In the case of cooperative prototyping, the cooperative work may be done in a physical third space that is neither the end-users' office nor the software developers' office (see "Sitings," previously). In the case of the delivery of iterated prototypes, each prototype is presented in the end-users' setting, but is unusual and only partially functional, and thus occasions reflection about its nature, its role in the end-users' work, and, ultimately, the work itself. In both cases, the invitation (or perhaps the necessity) of the end-users' actions to help shape the technology becomes an important means of refocusing their attention, as well as the attention of the software developers. The ensuing conversations are concerned with the interlinked feasibility of changes to technology and to work practices, with attributes of hybridity including polyvocal dialogues, challenging one another's assumptions, and developing plans for collective actions.

Claimed benefits. Some of the virtues of the low-tech prototyping approaches have also been claimed for the cooperative prototyping and "iterative delivery" approaches:

- **Enhanced communication and understanding** through grounding discussions in concrete artifacts.
- **Enhanced working relations** through a sense of shared ownership of the resulting design.

Additional claims for software-based prototypes include:

- **Earlier understanding of constraints** posed by the practical limitations of software.
- **Improved contextual grounding of the design** in the end-users' work practices.

CONCLUSION

My theme has been hybridity, and the ways in which selected methods in participatory design may bring useful attributes of hybridity or third space approaches into HCI work. I considered eight trends in PD—(a) selection of sites of shared work, (b) workshops, (c) stories, (d) end-user photography, (e) dramas, (f) creation of shared languages, (g) descriptive artifacts (low-tech prototypes), and (h) working prototypes—and I explored how each of these categories of practice may contribute to hybridity, and what advantages may result. The deliberate and selective use of hybridity has led to powerful methods in PD for increasing communication effectiveness, team coherence, innovation, and quality of outcome. Hybridity is thus at the heart of PD, fostering the critical discussions and reflections necessary to challenge assumptions and to create new knowledges, working practices, and technologies. When we consider HCI as a set of disciplines that lie between the space of work and the space of software development, we see that the hybrid third spaces developed within PD have much to offer HCI in general.

Table 54.2 summarizes the discussion of hybridity in PD, using the criteria derived from cultural studies (Table 54.1) and the experiences described in the eight areas of practice. Table 54.2 shows different patterns of hybridity for different methods, techniques, and practices.

Certain attributes are relatively common across practices, such as inbetweenness, questioning assumptions, negotiation, and heterogeneity as the norm. Other attributes are relatively rare, such as, considerations of what constitutes legitimate data for analysis or design, how those data are analyzed as evidence, and how conclusions are drawn in each of the several fields that are represented in a team. These are difficult questions in the study of disciplinarity (Chandler, Davidson, & Harootunian, 1994; Klein, 1996), so it is perhaps not surprising that there is relatively weak support for their exploration in participatory practices. For projects in which these are pivotal questions, we may need new methods that leverage hybridity in new ways. I hope that this survey of PD practices for creating third spaces will lead to new practices that strengthen these missing attributes. Conversely, I hope that new work in PD and HCI can help to ground some of the cultural studies discussions in new ways.

This chapter would not be complete without a list of unsolved problems in participatory design:

- **Participation by non-organized workforce.** The field of PD has long been concerned about how to engage in meaningful participative activities with workers or others who are not organized into a group with collective bargaining power or other collective representation (e.g., Greenbaum, 1993,

TABLE 54.2. Hybridity in Participatory Practices

Attribute	Sitings	Workshops	Stories	Photos	Dramas	Games	Language	Descriptive	Prototypes
Overlap/Inbetweenness	?	+	−	+	+	+	+	+	+
Marginality	+	+	−	?	+	+	?	+	?
Novelty	+	+	?	?	+	+	+	+	+
Uncertain/shared "ownership"	?	+	?	−	+	+	+	−	−
Selected attributes	+	?	+	+	−	+	+	−	+
Conflicts	+	+	+	−	+	−	+	−	+
Questioning assumptions	+	?	+	+	+	+	+	?	+
Mutual learning	+	+	+	+	+	+	+	?	+
Synthesis of new ideas	?	+	+	+	+	+	?	+	+
Negotiation/(co-)creation	+	+	+	+	+	+	+	+	+
Identities	−	−	+	+	−	?	?	+	?
Working language	−	?	+	+	−	+	+	+	+
Working assumptions and dynamics	+	?	+	+	+	+	+	?	+
Understandings	+	+	+	+	+	+	+	+	+
Relationships	?	+	+	+	−	+	?	+	?
Collective actions	?	+	?	+	?	?	?	+	+
Dialogues	+	+	+	+	+	+	+	+	+
Polyvocality	+	+	+	+	+	+	+	+	+
What is considered to be data?	−	−	−	+	−	−	+	+	−
What are the rules of evidence?	−	−	−	+	−	−	+	+	−
How are conclusions drawn?	−	−	−	?	−	−	+	−	−
↓ authority—↑ interpretation	+	?	+	+	+	+	+	?	+
↓ individualism—↑ collectivism	?	+	?	+	?	+	?	?	+
Heterogeneity as the norm	+	+	+	+	−	+	+	+	+

[a]Key: + practice includes this attribute of hybridity
− practice does not include this attribute
? not sure

1996; van den Besselaar, Greenbaum, & Mambrey, 1996). This has been a particularly difficult problem when we have tried to compare methods from one country (and political culture) to another (e.g., Muller et al., 1991).

- **Evaluation and metrics.** One of the weaknesses of the literature on participatory practices is the dearth of formal evaluations. There is a small set of papers that have examined software engineering projects across companies, and have found positive outcomes related to end-user participation (Cotton et al., 1988; Saarinen & Saaksjarvi, 1989). I have been unable to discover any formal experiments comparing participatory methods with nonparticipatory methods in a credible workplace context. Indeed, such studies would be difficult to perform, because they would require that a product be implemented and marketed twice (once with participation, and once without). The problem is made more difficult because measurements and metrics of organizational outcomes, user participation, and user satisfaction are currently vexing research issues (e.g., Garrety & Badham, 1998; Kappelman, 1995; for review, see Gasson, 1995).
- **Universal usability and "universal participation?"** Nearly all of the practices described in this chapter (and in the longer set of methods in Muller et al., 1997) are strongly visual and require hands-on manipulation of materials. These approaches violate the emerging requirements of universal usability for people with visual or motor disabilities (see, e.g., Universal Usability Fellows, 2000 and the *Proceedings of the Conference on Universal Usability*[16]; see also chapters in this volume by Vanderheiden [chapter] Czaja [chapter 39]; Marcus [chapter 18]; Newell, Carmichael, Gregor, Alm, & Waller [chapter 41]; Sears [chapter 42]; and Jacko, Leonard, & Scott [chapter 43]). In the previous edition of this book, I noted the irony that participatory design appeared to have failed in its inclusiveness with regard to people with disabilities. Happily, that problem is being addressed in at least three countries through the research and methodologies of Carmien, DePaula, Gorman, and Kintsch (2003); Davies, Marcella, McGrenere, and Purves (2004); Moffatt, McGrenere, Purves, and Klawe (2004); and Wu and colleagues (Wu, Richards, & Baecker,2004; Wu, Baecker, & Richards, 2005).

Note to Editors and Printers: Because I am citing many Scandinavian authors, I have alphabetized my references according to the Scandinavian alphabet: a, b, c. . . x, y, z, å, æ, ø.

References[17]

Anderson, W. L., & Crocca, W. T. (1993). Engineering practice and codevelopment of product prototypes. *Communications of the ACM, 36*(6), 49–56.

Anzaldúa, G. (1999). *La frontera/Borderlands*. San Francisco: Aunt Lute Books.

Bachmann-Medick, D. (1996). Cultural misunderstanding in translation: Multicultural coexistence and multicultural conceptions of world literature. *Erfurt Electronic Studies in English*, 7. Retrieved April 2, 2007, from http://webdoc.gwdg.de/edoc/ia/eese/artic96/bachmann/7_96.html

Beck, E. E. (1996). P for political? Some challenges to PD towards 2000. *PDC'96 Proceedings of the Participatory Design Conference* (pp. 117–125). Cambridge, MA: CPSR.

Beeson, I., & Miskelly, C. (2000). Dialogue and dissent in stories of community. *Proceedings of PDC 2000* (pp. 1–10). New York: CPSR.

Beeson, I., & Miskelly, C. (1998). Discovery and design in a community story. *Proceedings of PDC'98* (pp. 147–155). Seattle: CPSR.

Bertelsen, O. W. (1996). The Festival checklist: Design as the transformation of artifacts. *Proceedings of PDC 96* (pp. 93–102). Cambridge, MA: CPSR, 93–102.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems*. San Francisco: Morgan Kaufmann.

Bhabha, H. K. (1994). *The location of culture*. London: Routledge.

Binder, T. (1999). Setting the scene for improvised video scenarios. *Adjunct Proceedings of CHI'99* (pp. •••). Pittsburgh: ACM.

Bjerknes, G., & Bratteteig, T. (1995). User participation and democracy: A discussion of Scandinavian research on system development. *Scandinavian Journal of Information Systems, 7*(1), 73–98.

Bjerknes, G., Ehn, P., & Kyng, M. (Eds.). (1987). *Computers and democracy: A Scandinavian challenge.* Brookfield, VT: Gower.

Björgvinsson, E., & Hillgren, P. A-. (2004). On the spot experiments within healthcare. *Proceedings of PDC 2004* (pp. 93–101). Toronto: CPSR.

Blomberg, J., Giacomi, J., Mosher, A., &Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In D. Schuler & A. Namioka (Eds.), *Participatory design: Principles and practices*. Hillsdale, NJ: Erlbaum.

Boal, A. (1992). *Games for actors and non-actors* (A. Jackson, Trans.). London: Routledge. (Original work published 1974)

Bolton, R. (Ed.). (1989). *The contest of meaning: Critical histories of photography*. Cambridge, MA: MIT Press.

Braa, K. (1996). Influencing qualities of information systems—Future challenges for participatory design. *Proceedings of PDC'96* (pp. 15–24). Cambridge, MA: CPSR.

Brandt, E., & Grunnet, C. (2000). Evoking the future: Drama and props in user centered design. *Proceedings of PDC 2000* (pp. 11–20). New York: CPSR.

Brandt, E., & Messeter, J. (2004). Facilitating collaboration through design games. *Proceedings of PDC 2004* (pp. 121–131). Toronto: CPSR.

Briefs, U., Ciborra, C., & Schneider, L. (1983). *System design for, with, and by the users*. Amsterdam: North-Holland.

Bunch, C. (1987). Passionate Politics: Essays 1968–1986: Feminist theory in action. New York: St. Martin's Press.

Buur, J., Binder, T., & Brandt, E. (2000). Taking video beyond "hard data" in user centred design. *Proceedings of PDC 2000* (pp. 21–29). New York: CPSR.

Buur, J., & Bødker, S. (2000). From usability lab to "design collaboratorium": Reframing usability practice. *Proceedings of DIS'00* (pp. 297–307). New York: ACM.

[16]Available through ACM, www.acm.org.

[17]For a more general PD bibliography, see http://www.cpsr.org/prevsite/program/workplace/PD-history.html

Buzan, T., & Buzan, B. (1996). *The mind map book: How to use radiant thinking to maximize your brain's untapped potential*. New York: Plume.

Bødker, S., & Buur, J. (2002). The design collaboratorium—A place for usability design. *Transactions on Computer-Human Interaction, 19*(2), 152–169.

Bødker, K., Kensing, F., & Simonsen, J. (2004). *Participatory IT design: Designing for business and workplace realities*. Cambridge, MA: MIT Press.

Bødker, S. (1990). *Through the interface: A human activity approach to user interface design.* Hillsdale, NJ: Erlbaum.

Bødker, S., Ehn, P., Kyng, M., Kammersgaard, J., & Sundblad, Y. (1987): A UTOPIAN experience: On design of powerful computer-based tools for skilled graphic workers. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge*. Brookfield, VT: Gower.

Bødker, S., & Grønbæk, K. (1991). Design in action: From prototyping by demonstration to cooperative prototyping. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems*. Hillsdale: NJ: Erlbaum.

Bødker, S., Grønbæk, K., & Kyng, M. (1993). Cooperative design: Techniques and experiences from the Scandinavian scene. In D. Schuler & A. Namioka (Eds.), *Participatory design: Principles and practices*. Hillsdale, NJ: Erlbaum.

Bødker, S., Knudsen, J. L., Kyng, M., Ehn, P., & Madsen, K. H. (1988). Computer support for cooperative design. *CSCW'88: Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 377–394). Portland, OR: ACM.

Cameron, M. (1998). Design for safety: Working with residents to enhance community livability. *Proceedings of PDC'98* (pp. 226–227). Seattle: CPSR.

Carmien, S., DePaula, R., Gorman, A., & Kintsch, A. (2003). Increasing workplace independence for people with cognitive disabilities by leveraging distributed cognition among caregivers and clients. *Proceedings of GROUP'03* (pp. 95–104). Sanibel Island, FL: ACM.

Carrillo, R. (2000). Intersections of official script and learners' script in *Third Space*: A case study on Latino families in an after-school computer program. *Proceedings of Fourth International Conference of the Learning Sciences* (pp. 312–313). Mahwah, NJ: Erlbaum.

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.

Carroll, J. (Ed.). (1995). *Scenario-based design for human-computer interaction*. New York: Wiley.

Carter, S., & Mankoff, J. (2005). When participants do the capturing: The role of media in diary studies. *Proceedings of CHI 2005* (pp. 899–908). Portland, OR: ACM.

Chandler, J., Davidson, A. I., & Harootunian, H. (Eds.). (1994). *Questions of evidence: Proof, practice, and persuasion across the disciplines*. Chicago: University of Chicago Press.

Checkland, P. (1981). *Systems thinking, systems practice*. New York: Wiley.

Chin, G., Schuchardt, K., Myers, J., & Gracio, D. (2000). Participatory workflow analysis: Unveiling scientific research processes with physical scientists. *Proceedings of PDC 2000* (pp. 30–39). New York: CPSR.

Clement, A., Kolm, P., & Wagner, I. (Eds.) (1994). *NetWORKing: Connecting workers in and between organizations*. IFIP Transactions A-38. Amsterdam: North-Holland.

Cotton, J. L., Vollrath, D. A., Froggatt, K.L, Lengnick-Hall, M. L., & Jennings, K. R. (1988). Employee participation: Diverse forms and different outcomes. *Academy of Management Review 13*, 8–22.

Crabtree, A. (1998). Ethnography in participatory design. *Proceedings of PDC 98* (pp. 93–105). Seattle: CPSR.

Daly-Jones, O., Bevan, N., & Thomas, C. (1999). *IN-USE 6.2 Handbook of user-centred design*. Teddington, United Kingdom: Serco Usability Services.

Dandavate, U., Steiner, D., & William, C. (2000). Working anywhere: Co-Design through participation. *Proceedings of CoDesigning 2000* (pp. 101–110). London: Springer.

Davies, R., Marcella, S, McGrenere, J., & Purves, B. (2004). The ethnographically informed participatory design of a PDA to support communication. *Proceedings of ASSETS'04* (pp. 153–160). Atlanta, GA: ACM.

Dingawaney, A., & Maier, C. (1994). *Between languages and cultures: Translation and cross-cultural texts*. Pittsburgh, PA: University of Pittsburgh Press.

Docherty, P., Fuchs-Kittowski, K., Kolm, P., & Matthiessen, L. (1987). *System design for human development and productivity: Participation and beyond*. Amsterdam: North-Holland.

Dray, S. M. (1992, October). Understanding and supporting successful group work in software design: Lessons from IDS [Position paper]. In J. Karat & J. Bennett (Chairs), *Understanding and supporting successful group work in software design.* Workshop conducted at the CSCW '92 conference, Toronto.

Druin, A. (1999). Cooperative inquiry: Developing new technologies for children with children. *Proceedings of CHI'99* (pp. 592–599). Pittsburgh: ACM.

Druin, A., Alborzi, H., Boltman, A., Cobb, S., Montemayor, J., Neale, H., et al. (2000). Participatory design with children: Techniques, challenges, and successes. *Proceedings of PDC 2000* (pp. 226–227). New York: CPSR.

Dykstra, E. A., & Carasik, R. P. (1991). Structure and support in cooperative environments: The Amsterdam Conversation Environment. *International Journal of Man-Machine Studies, 34*, 419–434.

Ehn, P. (1993). Scandinavian design: On participation and skills. In P. S. Adler & T. A. Winograd (Eds.), *Usability: Turning technologies into tools*. New York: Oxford University Press.

Ehn, P. (1988). *Work-oriented design of computer artifacts*. Falköping, Sweden: Arbetslivcentrum/Almqvist and Wiksell International.

Ehn, P., & Kyng, M. (1987). The collective resource approach to systems design. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge*. Brookfield, VT: Gower.

Ehn, P., & Kyng, M. (1991). Cardboard computers: Mocking-it-up or hands-on the future. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Ehn, P., & Sjögren, D. (1991). From system descriptions to scripts for action. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Ehn, P., & Sjögren, D. (1986). Typographers and carpenters as designers. In *Proceedings of skill-based automation*. Karlsruhe, Germany.

Erickson, T. (1996). Design as story-telling. *Interactions*, July/August 1996.

Evanoff, R. (2000). The concept of "third cultures" in intercultural ethics. *Eubios Journal of Asian and International Bioethics, 10*, 126–129.

Fanderclai, T. (1996). Like magic, only real. In L. Cherny & E.R. Weise (Eds.), *Wired women: Gender and new realities in cyberspace.* Seattle: Seal Press.

Fanderclai, T. (1995). MUDs in education: New environments, new pedagogies. *Computer-Mediated Communication, 2*(1), 8.

Floyd, C. (1987). Outline of a paradigm change in software engineering. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge.* Brookfield, VT: Gower.

Floyd, C. (1993). STEPS—A methodical approach to PD. *Communications of the ACM, 36I*(6), 83.

Floyd, C., Mehl, W.-M., Reisin, F.-M., Schmidt, G., & Wolf, G. (1989). Out of Scandinavia: Alternative approaches to software design and system development. *Human-Computer Interaction, 4*(4), 253–350.

Fowles, R. A. (2000). Symmetry in design participation in the built environment: Experiences and insights from education and practice. *Proceedings of CoDesigning 2000* (pp. 59–70). London: Springer.

Garrety, K., & Badham, R. (1998, June). *The four-dimensional politics of technology, or postmodernising participatory design*. Presented at the Cultural Politics of Technology workshop, Centre for Technology and Society, Trondheim. http://www.ntnu.no/sts/content/Papers/Four.html

Gasson, S. (1995). User involvement in decision-making in information systems development. *Proceedings of 18thth IRIS*. Gjern Denmark: IRIS Association.

Gärtner, J., & Wagner, I. (1995). *Political frameworks of systems design from a cross-cultural perspective*. IFIP WG.9.1 Workshop. Aarhus: IFIP.

Gjersvik, R., & Hepsø, V. (1998). Using models of work practice as reflective and communicative devices: Two cases from the Norwegian offshore industry. *Proceedings of PDC'98* (pp. 107–116). Seattle: CPSR.

Greenbaum, J. (1993). A design of one's own: Towards participatory design in the United States. In D. Schuler & A. Namioka (Eds.), *Participatory design: Principles and practices*. Hillsdale, NJ: Erlbaum.

Greenbaum, J. (1996). Post modern times: Participation beyond the workplace. *Proceedings of PDC'96* (pp. 65–72). Cambridge, MA: CPSR.

Greenbaum, J., & Kyng, M. (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Grenfell, M. (1998,). Border-crossing: Cultural hybridity and the rural and small schools practicum. Paper presented at the annual conference of the Australian Association for Research in Education, Adelaide, November–December 1998.

Gruen, D. (2001). *Stories in design tutorial*. IBM Make It Easy Conference. Cambridge, MA, USA, June 2001.

Gruen, D. (2000). *Storyboarding for design: An overview of the process.* Cambridge, MA: Lotus Research. http://www.research.ibm.com/cambridge, under "Papers".

Grønbæk, K. (1989). Extending the boundaries of prototyping—Toward cooperative prototyping. *Proceedings of 12thth IRIS* (pp. 219–238). Aarhus: IRIS Association.

Henderson, A., & Kyng, M. (1991). There's no place like home: Continuing design in use. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Holmström, J. (1995). The power of knowledge and the knowledge of power: On the systems designer as a translator of rationalities. *Proceedings of the 18thth IRIS*. Göteborg: IRIS Association.

Horgan, T., Joroff, M. L., Porter, W. L., & Schön, D. A. (1998). *Excellence by design— Transforming workplace and work practice*. New York: Wiley.

Howard, S., Carroll, J., Murphy, J., & Peck, J. (2002). Using 'endowed props' in scenario-based design. *Proceedings of NORDCHI 2002* (pp. 1–10). Aarhus: ACM, New York.

Hulkko, S., Mattelmäki, T., Virtanen, K., & Keinonen, T. (2004). Mobile probes. *Proceedings of NORDCHI 2004* (pp. 43–51). Tampere: ACM.

Hultcrantz, J., & Ibrahim, A. (2002). Contextual workshops: User participation in the evaluation of future concepts. *Proceedings of PDC 2002* (pp. 344–348). Malmö: CPSR.

Iacucci, G., Iacucci, C., & Kuutti, K. (2002). Imaging and experiencing in design—The role of performances. *Proceedings of NORCHI 2002* Aarhus: October 2002.

Iacucci, G., & Kuutti, K. (2002). Everyday life as a stage in creating and performing scenarios for wireless devices. *Personal and Ubiquitous Computing, 6*, 299–306.

Irestig, M., & Timpka, T. (2002). Dynamic interactive scenario creation: A method for extending participatory design to large system development projects. *Proceedings of PDC 2002* (pp. 317–322). Malmö: CPSR.

Isomursu, M., Kuutti, K., & Väinämö, S. (2004). Experience clip: Method for user participation and evaluation of mobile concepts. *Proceedings of PDC 2004* (pp. 83–92). Toronto: CPSR.

Iversen, O. S., & Buur, J. (2002). Design is a game: Developing design competence in a game setting. *Proceedings of PDC 2002* (pp. 22–28). Malmö: CPSR.

Johansson, M., Fröst, P., Brandt, E., Binder, T., & Messeter, J. (2002). Partner engaged design: New challenges for workplace design. *Proceedings of PDC 2002*. Malmö: CPSR.

Jungk, R., & Mullert, N. (1987). *Future workshops: How to create a desirable future.* London: Institute of Social Invention.

Kaindl, H., Constantine, L., Karat, J., & Muller, M. J. (2001). Methods and modeling: Fiction or useful reality? (panel). *CHI 2001 Extended Abstracts* (pp. 213–214). Seattle: ACM.

Kappelman, L. (1995). Measuring user involvement: A diffusion of innovation perspective. *The DATABASE for Advances in Information Systems, 26*(2&3), 65–86.

Karasti, H., Baker, K. S., & Bowker, G. C. (2002). Ecological storytelling and collaborative scientific activities. *SIGGROUP Bulletin, 2* (2), 29–30.

Karttunen, F. (1994). *Between worlds: Interpreters, guides, and survivors*. New Brunswick, NJ: Rutgers University Press.

Kensing, F., & Blomberg, J. (Eds.). (1998). *Computer Supported Cooperative Work, 7*(3–4).

Kensing, F., & Madsen, K. H. (1991). Generating visions: Future workshops and metaphorical design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems.* Hillsdale, NJ: Erlbaum.

Kensing, F., & Munk-Madsen, A. (1993). PD: Structure in the toolbox. *Communications of the ACM, 36*(6), 78–85.

Kensing, F., Simonsen, J., & Bødker, K. (1996). MUST—A method for participatory design. *Proceedings of PDC'96* (pp. 129–140). Cambridge, MA: CPSR.

Klein, J. T. (1996). *Crossing boundaries: Knowledge, disciplinarities, and interdisciplinarities*. Charlotteville, NC: University Press of Virginia.

Klær, A., & Madsen, K. H. (1995). Participatory analysis of flexibility. *Communications of the ACM, 38*(5), 53–60.

Krabbel, A., & Wetzel, I. (1998). The customization process for organizational package information systems: A challenge for participatory design. *Proceedings of PDC 98* (pp. 45–54). Seattle: CPSR.

Kuutti, K., Iacucci, G., & Iacucci, C. (2002). Acting to know: Improving creativity in the design of mobile services by using performances. *Proceedings of C&C 02* (pp. 95–102). Loughborough: ACM.

Kwok, J. Y.-C. (2004). The weight of space: Participatory design research for configuring habitable space for new arrival women in Hong Kong. *Proceedings of PDC 2004* (pp. 183–192). Toronto: CPSR.

Kyng, M., & Matthiessen, L. (Eds.). (1997). *Computers in design and context*. Cambridge, MA: MIT Press.

Lafreniére, D. (1996). CUTA: A simple, practical, and low-cost approach to task analysis. *Interactions, 3*(5), 35–39.

Lanzara, G. F. (1983). The design process: Frames, metaphors, and games. In U. Briefs, C. Ciborra, & L. Schneider (Eds.), *Systems design for, with, and by the users*. Amsterdam: North-Holland.

Levinger, D. (1998). Participatory design history. Retrieved April 2, 2007, from http://www.cpsr.org/prevsite/conferences/pdc98/history.html

Luck, R. (2000). Does "inclusive design" require an inclusive design process? *Proceedings of CoDesigning 2000* (pp. 71–79). London: Springer.

Lyotard, J.-F. (1984). *The post-modern condition: A report on knowledge*. Minneapolis: University of Minnesota Press.

MacLean, A., Carter, K., Lovstrand, L., & Moran, T. (1990). User-tailorable systems: Pressing the issues with buttons. *Proceedings of CHI '90* (pp. 175–182). Seattle: ACM.

Madsen, K. H. (1999). *Communications of the ACM* AU: Please provide page numbers for this source. AU: Please provide page numbers for this source, *42*(5), •••.

Maher, M. L., Simoff, S J., & Gabriel, G. C. (2000). Participatory design and communication in virtual environments. *Proceedings of PDC 2000* (pp. 117–134). New York: CPSR.

Mattelmäki, T., & Batarbee, K. (2002). Empathy probes. *Proceedings of PDC 2002* (pp. 266–271). Malmö: CPSR.

McLagan, P., & Nel, C. (1995). *The age of participation: New governance for the workplace and the world*. San Francisco: Berrett-Koehler.

McPhail, B., Costantino, T., Bruckmann, D., Barclay, R., & Clement, A. (1998). CAVEAT exemplar: Participatory design in a non-profit volunteer organization. *Computer Supported Cooperative Work, 7*, 223–241.

Merkel, C. B., Xiao, L., Farooq, U., Ganoe, C. H., Lee, R., Carroll, J. M., et al. (2004). Participatory design in community computing contexts: Tales from the field. *Proceedings of PDC 2004* (pp. 1–10). Toronto: CPSR.

Moffatt, K., McGrenere, J., Purves, B., & Klawe, M. (2004). The participatory design of a sound an image enhanced daily planner for people with aphasia. *Proceedings of CHI 2004* (pp. 407–414). Portland, OR: ACM.

Monk, A., & Howard, S. (1998). The rich picture: A tool for reasoning about work context. *Interactions, 2*, 21–30.

Mogensen, P., & Trigg, R. (1992). Artifacts as triggers for participatory analysis. *Proceedings of PDC'92* (pp. 55–62). Cambridge MA USA: CPSR.

Muller, M. J. (2001). Layered participatory analysis: New development in the CARD technique. *Proceedings of CHI 2001* (pp. 90–97). Seattle: ACM.

Muller, M. J. (1999a). *Catalogue of scenario-based methods and methodologies*. Lotus Research Technical Report 99-06. Retrieved April 2, 2007, from http://www.research.ibm.com/cambridge

Muller, M. J. (1999b). Translation in HCI: Toward a research agenda. Lotus Research Technical Report 99–05. Retrieved April 2, 2007, from http://www.research.ibm.com/cambridge

Muller, M. J. (1997a). Ethnocritical heuristics for reflecting on work with users and other interested parties. In M. Kyng & L. Matthiessen (Eds.), *Computers in context and design*. Cambridge, MA: MIT Press.

Muller, M. J. (1997b). Translation in HCI: Formal representations for work analysis and collaboration. In *Proceedings of CHI'97* (pp. 544–545). Atlanta, GA: ACM.

Muller, M. J., Blomberg, J. L., Carter, K., Dykstra, E. A., Greenbaum, J., & Halskov Madsen, K. (1991). Panel: Participatory design in Britain and North America: Responses to the "Scandinavian challenge." *Proceedings of CHI'91* (pp. 389–392). New Orleans: ACM.

Muller, M. J., Carr, R., Ashworth, C. A., Diekmann, B., Wharton, C., Eickstaedt, C., et al. (1995a). Telephone operators as knowledge workers: Consultants who meet customer needs. *Proceedings of CHI'95* (pp. 130–137). Denver, CO: ACM.

Muller, M. J., Hallewell Haslwanter, J. D., & Dayton, T. (1997). Participatory practices in the software lifecycle. In M. Helander, T. Landauer, and Prabhu, P. (Ed.), *Handbook of human-computer interaction*. Amsterdam: Elsevier.

Muller, M. J., & Kuhn, S. (Eds.). (1993). *Communications of the ACM*, *36*(6).

Muller, M. J., Tudor, L. G., Wildman, D. M., White, E. A., Root, R. W., Dayton, T., et al. (1995b). Bifocal tools for scenarios and representations in participatory activities with users. In J. Carroll (Ed.), *Scenario-based design for human-computer interaction*. New York: Wiley.

Muller, M. J., White, E. A., & Wildman, D. M. (1993). Taxonomy of PD practices: A brief practitioner's guide. *Communications of the ACM, 36*(6), 26–28.

Muller, M. J., Wildman, D. M., & White, E. A. (1994). *Participatory design through games and other group exercises*. Tutorial at CHI'94 conference, Boston, MA.

Mumford, E. (1983). *Designing human systems for new technology: The ETHICS method.* Manchester, United Kingdom: Manchester Business School.

Mumford, E., & Henshall, D. (1979/1983). *Designing participatively: A participative approach to computer systems design*. Sandbach, United Kingdom: Manchester Business School.

Mørch, A. I., Engen, B. K., & Åsand, H.-R. H. (2004). The workplace as a learning laboratory: The winding road to e-learning in a Norwegian service company. *Proceedings of PDC 2004* (pp. 142–151). Toronto: CPSR.

Nisonen, E. (1994). Women's safety audit guide: An action plan and a grass roots community development tool. *CPSR Newsletter, 12,* (3) (p. 7). Palo Alto, CA, USA.

Noble, A., & Robinson, C. (2000). For the love of the people: Participatory design in a community context. *Proceedings of CoDesigning 2000* (pp. 81–91). London: Springer.

Nordichi. (2006). Nordichi.org. Retrieved April 2, 2007, from http://www.nordichi.org

Noro, K., & Imada, A. S. (Eds.). (1991). *Participatory ergonomics*. London: Taylor and Francis.

Nygaard, K. (1975). Kunnskaps-strategi for fagbevegelsen (Knowledge strategy for trade unions). *Nordisk Forum, 6, 10*(2), 15–27.

Orr, J., & Crowfoot, N. C. (1992). Design by anecdote—The use of ethnography to guide the application of technology to practice. *PDC'92: Proceedings of the Participatory Design Conference* (pp. 31–37). Cambridge, MA: CPSR.

Patton, J. W. (2000). Picturing commutes: Informant photography and urban design. *Proceedings of PDC 2000* (pp. 318–320). New York: CPSR.

Pedell, S. (2004). Picture scenarios: An extended scenario-based method for mobile appliance design. *Proceedings of OZCHI* 2004 (pp. 17–18). Wollongong, Australia.

Pedersen, J., & Buur, J. (2000). Games and moves: towards innovative codesign with users. *Proceedings of CoDesigning 2000* (pp. 93–100). London: Springer.

Reid, F. J. M., & Reed, S. E. (2000). Interaction and entrainment in collaborative design meetings. *Proceedings of CoDesigning 2000* (pp. 233–241). London: Springer.

Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM, 37*(4), 21–27.

Robertson, T. (1996). Participatory design and participative practices in small companies. *Proceedings of PDC'96* (pp. 35–43). Cambridge MA USA: CPSR.

Robertson, T. (1998). Shoppers and tailors: Participative practices in small Australian design companies. *Computer Supported Cooperative Work, 7*(3–4), 205–221.

Robins, J. (1999). Participatory design (class notes). Champaign, IL: University of Illinois. Retrieved April 2, 2007, from http://www.lis.uiuc.edu/~jrobins/pd/

Saarinen, T., & Saaksjarvi, M. (1989). The missing concepts of user participation: An empirical assessment of user participation and information system success. In *Proceedings of the 12th IRIS* (Information System Research in Scandinavia) (pp. 533–553). Aarhus, Denmark: Aarhus University

Salvador, T., & Howells, K. (1998). Focus Troupe: Using drama to create common context for new product concept end-user evaluations. *Proceedings of CHI'98* (pp. 251–252) Los Angeles: ACM.

Salvador, T., & Sato, S. (1998). Focus troupe: Mini-workshop on using drama to create common context for new product concept end-user evaluations. *Proceedings of PDC'98* (pp. 197–198). Seattle: CPSR.

Salvador, T., & Sato, S. (1999). Methods tools: Playacting and focus troupes: theater techniques for creating quick, intense, immersive, and engaging focus group sessions. *Interactions, 6*(5), 35–41.

Sanders, E. B.-N. (2000). Generative tools for co-designing. *Proceedings of CoDesigning 2000* (pp. 3–12). London: Springer.

Sanders, E. B.-N., & Branaghan, R. J. (1998). Participatory expression through image collaging: A learning-by-doing experience. *Proceedings of PDC'98* (p. 199). Seattle: CPSR.

Sanders, E. B.-N., & Nutter, E. H. (1994). Velcro-modeling and projective expression: Participatory design methods for product development. *PDC'94: Proceedings of the Participatory Design Conference* (p. 125). Chapel Hill, NC: CPSR.

Scrivener, S. A. R., Ball, L. J., & Woodcock, A. (2000). *Collaborative design: Proceedings of Co-Designing 2000*. London: Springer.

Schuler, D., & Namioka, A. (1993). (Eds.). *Participatory design: Principles and practices*. Hillsdale, NJ: Erlbaum.

Segall, P., & Snelling, L. (1996). Achieving worker participation in technological change: The case of the flashing cursor. In *Proceedings of PDC'96*. Cambridge MA USA: CPSR.

Slater, J. (1998). Professional misinterpretation: What is participatory design? *Proceedings of PDC'98* (pp. 83–92). Seattle: CPSR.

Star, S. L., & Griesemer, J. R. (1989). Institutional ecology, "translations," and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907–1939. *Social Studies of Science, 19*, 387–420.

Suchman, L. (2002). Located accountabilities in technology production. *Scandinavian Journal of Information Systems, 14*(2), 91–105.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication*. Trowbridge, Wiltshire: Cambridge University Press.

Suchman, L., & Trigg, R. (1991). Understanding practice: Video as a medium for reflection and design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Svanæs, D., & Seland, G. (2004). Putting the users center stage: Role playing and low-fi prototyping enable end users to design mobile systems. *Proceedings of CHI 2004* (pp. •••). Vienna, Austria: ACM.

Thackara, J. (2000). Edge effects: The design challenge of pervasive interface. Plenary presentation at CHI 2000, April 2000.

Trigg, R. H. (2000). From sandbox to "fundbox": Weaving participatory design into the fabric of a busy non-profit. *Proceedings of PDC 2000* (pp. 174–183). New York: CPSR.

Tscheligi, M., Houde, S., Marcus, A., Mullet, K., Muller, M. J., & Kolli, R. (1995). Creative prototyping tools: What interaction designers really need to produce advanced user interface concepts. *CHI'95 conference companion* (pp. 170–171). Denver: ACM.

Tschudy, M. W., Dykstra-Erickson, E. A., & Holloway, M. S. (1996). PictureCARD: A storytelling tool for task analysis. *PDC'96 Proceedings of the Participatory Design Conference* (pp. 183–191). Cambridge, MA: CPSR.

Tudor, L. G., Muller, M. J., Dayton, T.,& Root, R. W. (1993). A participatory design technique for high-level task analysis, critique, and redesign: The CARD method. *Proceedings of the HFES'93* (pp. 295–299). Seattle: Human Factors and Ergonomics Society.

Törpel, B., & Poschen, M. (2002). Improving infrastructures by transforming narratives. *Proceedings of PDC 2002*. Malmö: CPSR.

Universal Usability Fellows. (2000). •••. Retrieved April 2, 2007 from http//www.universalusability.org/

van den Besselaar, P., Clement, A., & Jaervinen, P. (1991). *Information system, work and organization design*. Amsterdam: North-Holland.

van den Besselaar, P., Greenbaum, J., & Mambrey, P. (1996). Unemployment by design: Participatory design and the changing structure of the workforce in the information society. *Proceedings of PDC 96* (pp. 199–200). Cambridge, MA: CPSR.

Winner, L. (1994). Political artifacts in Scandinavia: An American perspective. *Scandinavian Journal of Information Systems, 6*(2), 85–94.

Wixon, D., & Ramey, J. (Eds.) (1996). *Field methods casebook for software design*. New York: Wiley.

Wu, M., Baecker, R., & Richards, B. (2005). Participatory design of an orientation aid for amnesics. *Proceedings of CHI 2005* (pp. 511–520). Portland, OR, USA: ACM.

Wu, M., Richard, B., & Baecker, R. (2004). Participatory design with individuals who have amnesia. *Proceedings of PDC 2004* (pp. 214–223). Toronto: CPSR.

•55•

UNIFIED USER INTERFACE DEVELOPMENT: NEW CHALLENGES AND OPPORTUNITIES

Anthony Savidis and Constantine Stephanidis
Univeristy of Crete

Introduction **1084**
Automatic User Interface Adaptation 1084
The Concept of Unified User Interfaces 1084
Unified User Interface Design **1084**
The Design Problem 1084
Polymorphic Task Hierarchies 1085
The Conduct of the Method 1086
Transitions from the abstract task design state 1087
Transitions from the alternative subhierarchies design state 1087
Transitions from the task hierarchy decomposition state 1087
Transitions from the physical task design state 1087
Relationships Among Alternative Styles 1087
Exclusion 1087
Compatibility 1088
Substitution 1088
Augmentation 1088
Unified Interface Engineering **1088**
Architectural Overview 1089
User-information server 1089
Context parameters server 1091
Decision-making component 1091
Dialogue patterns component 1092
Adaptability and Adaptivity Cycles 1093
Dynamic User Interface Assembly 1093
Polymorphic Containment Hierarchies 1095
Decision-Making Specification 1096
Unified Development in Ambient Intelligence **1096**
Ambient Interactions 1097
Unified Ambient User Interfaces 1097
Remote Wireless GUI Service 1100
Test Case Applications 1101
Abstract Interaction Object Library and Polymorphic Alternatives 1102
Conclusions **1103**
Acknowledgments **1104**
References **1105**

INTRODUCTION

This chapter discusses the unified user interface development paradigm as an effective software engineering recipe for user interfaces that can be automatically adapted to individual end users and contexts of use. The distinctive procedural, representational, and programming properties of this development discipline are highlighted, qualifying it as a plausible and cost-effective approach towards the goal of developing automatic user interface personalization. The method conveys a fundamentally new perspective into the development of user interfaces, emphasizing a principled, systematic and evolutionary approach towards coping with diversity, by progressively encapsulating and coordinating in an extensible development structure all alternative interaction artifacts. Finally, the consolidated experience in applying the unified user interface development paradigm in the emerging field of ambient intelligence is discussed, by explaining the way the original method is effectively applied to build applications with unified ambient interfaces.

Automatic User Interface Adaptation

The notion of automatic user interface adaptation reflects the capability of interactive software to adapt during runtime to the individual end user, as well as to the particular context of use, by delivering the most appropriate interactive experience. The storage location, origin, and format of user-oriented information may vary. For example, information may be stored in profiles, indexed by unique user identifiers, extracted from user-owned cards, entered by the user in an initial interaction session, or inferred by the system through continuous interaction monitoring and analysis. Additionally, usage-context information, such as user location, environment noise, network bandwidth, and so forth, is normally provided by special purpose equipment, like sensors, or system-level software. In order to support optimal interface delivery for individual user- and usage-context-attributes, it is required that for any given user task or group of user activities, the implementations of the alternative best fit interface components are either implementationally encapsulated or appropriately locatable (e.g., remote components, downloadable plug-ins, dynamically linked libraries, etc.).

The Concept of Unified User Interfaces

A unified user interface is the interaction-specific software of software applications or services, which is capable of self-adapting to the characteristics of the individual end user and context of use. Such an adaptation may reflect varying patterns of interactive behavior, at the physical, syntactic, or semantic level of interaction, to accommodate specific user- and context-oriented parameters. Practically speaking, from the end-user point of view, a unified user interface is actually an interface that can automatically adapt to the individual user attributes (e.g., requirements, abilities, and preferences), as well as to the particular characteristics of the usage-context (e.g., computing platform, peripheral devices, interaction technology, and surrounding environment). Therefore, a unified user interface realizes the combination of

- User-adapted behavior, such as the automatic delivery of the most appropriate user interface for the particular end user (requires user awareness)
- Usage-context adapted behavior, such as the automatic delivery of the most appropriate user interface for the particular situation of use (requires usage context awareness)
- Hence, the characterization unified does not have any particular behavioral connotation, at least as seen from an end-user perspective. Instead, the notion of unification reflects the specific software engineering strategy needed to accomplish this behavior, emphasizing the proposed systematic development-oriented discipline. More specifically, in order to realize this form of adapted behavior, a unified user interface reflects the following fundamental development properties.
- It encapsulates alternative dialogue patterns (e.g., implemented dialogue artifacts), for various dialogue design contexts (e.g., a subtask, a primitive user action, a visualization), appropriately associated to the different values of user- and usage-context related attributes. The need for such alternative dialogue patterns may only be dictated by the user interface design process, when, given any particular design context, for different user- and usage-context attribute values, alternative design artifacts are needed to accomplish optimal interaction
- It implementationally encapsulates representation schemes for user- and usage-context parameters, internally utilizing user- and usage-context information resources (e.g., repositories, servers), to extract or to update user- and usage-context information.
- It encapsulates the necessary user interface design knowledge and decision-making capability for activating, during runtime, the most appropriate dialogue patterns (e.g., interactive software components), according to particular instances of user- and usage-context attributes.

The distinctive property of unified user interfaces to encapsulate alternative, mutually exclusive, user interface design artifacts, each purposefully designed and implemented as an optimal alternative for its corresponding target attributes of the user- and usage-context models, assuming a particular design context, constitutes one of the main contributions of this research work within the user interface software engineering field.

UNIFIED USER INTERFACE DESIGN

The Design Problem

Universal design in HCI reflects the principle to address potentially all users and usage contexts—anyone, anyplace, anytime. Its main objective is to ensure that each end user is given the most appropriate interactive experience, by supporting both accessible and high-quality interaction. In this context, in order to accommodate genuinely universal design in the production process of software application and services, two key issues need

to be optimally addressed. The first is scientific, primarily concerning the way the particular problem is to be technically resolved, while the second is cost specific, reflecting the criticality for cost-effective and economically viable solutions. Clearly, producing and enumerating distinct user interface designs through the conduct of multiple design processes is an impractical solution, since the overall cost for managing in parallel such a large number of independent design processes, as well as for transforming each produced user interface version into a target software implementation, would be unacceptable both for the design and the software implementation phases. Instead, a design process is required which may lead to a single design outcome that appropriately links and organizes the differentiating aspects of the resulting interactive application, around common abstract design structures and patterns, making it far easier to (a) map to a target software system implementation, and (b) maintain, update, and extend the design itself.

The need for introducing alternative design artifacts, even for the same specific design context (such as a particular subtask), emerges from the fact that, in universal design, the design problem encompasses largely varying parameters, such as user- and usage-context parameters. Consequently, when designing for any particular dialogue design context, it is likely that differentiating values of those problem parameters dictate the design of diverse dialogue artifacts. This issue introduces two important requirements for pursuing a suitable design method.

The first is that such a method should offer the capability to associate multiple alternative dialogue artifacts to a particular single design context, due to the varying design problem parameters, by enabling the unambiguous association of each alternative artifact with its corresponding values of the problem parameters.

The second is that the method should emphasize capturing of the more abstract structures and patterns inherent within user interface designs, enabling the hierarchical incremental specialization towards the lower physical level of interaction, making it possible to introduce alternative dialogues patterns as close to the physical design as possible. This will make it far easier for the design space to be updated and to evolve, since modifications and extensions due to the consideration of additional values of the problem parameters (e.g., considering new user- and usage-context attribute values) can be applied locally closer to the lower levels of the design, without affecting the rest of the design space.

To demonstrate briefly the need for supporting alternative dialogue artifacts for the same design context, an example from a real-life application will be used. The AVANTI web browser (Stephanidis, Paramythis, Sfyrakis, & Savidis, 2001) was developed to enable web access by supporting adaptation to the particular user as well as to the context of use. During the user interface design phase, while concentrating on the design context of the link dialogue task, alternative designs have been dictated, due to varying user- and usage-context parameters considered, as shown in Fig. 55.1.

Since the differing artifacts have been part of the final AVANTI web browser user interface design, the design representation formalism needed to enable their copresence within the resulting design space, by clearly associating each artifact to the link-selection task and its corresponding values of the

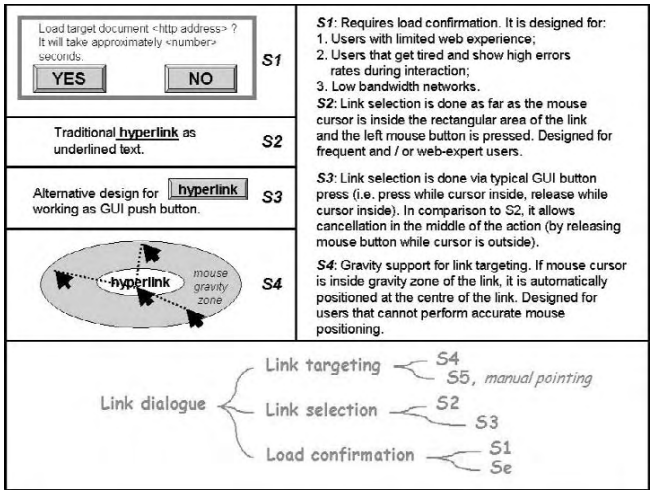


FIGURE 55.1. Designing alternative artifacts for the Link dialogue task. Se is used to indicate "empty" (i.e. no load confirmation dialogue supported). S5 is the typical manual link targeting GUI dialogue through the mouse.

user- and usage-context parameters. A loose design notation is employed in Fig. 55.1, to show hierarchical task analysis (subtask sequencing is omitted for clarity), as well as the need for alternative incarnations of a single task (e.g., styles S2/S3 for link selection, styles S1/Se for load confirmation, and styles S4/S5 for link targeting). Following this example taken from the AVANTI web browser, during runtime, depending on the particular end user and usage context attribute values, the appropriate corresponding implemented artifacts will only have to be activated.

Polymorphic Task Hierarchies

A polymorphic task hierarchy combines three fundamental properties: (a) hierarchical decomposition, (b) polymorphism, and (c) task operators. The hierarchical decomposition adopts the original properties of hierarchical task analysis for incremental decomposition of user tasks to lower level actions. The polymorphism property provides the design differentiation capability at any level of the task hierarchy, according to particular user- and usage-context attribute values. Finally, task operators, which are based on the communicating sequential processes (CSP) language for describing the behavior of reactive systems (Hoare, 1978), enable the expression of dialogue control flow formulae for task accomplishment. Those specific operators, taken from the domain of reactive systems and process synchronization, have been selected due to their appropriateness in expressing temporal relationships of user actions and tasks. However, designers may freely employ additional operators as needed (e.g., the set is not closed) or may choose to document dialogue sequencing and control outside the task-structure in natural language, when it engages more comprehensive algorithmic logic (e.g., consider the verbally documented precondition if the logged user is a guest, no sign-in is required, else, the access privileges should be checked and the sign-in dialogue is activated before chat).

The concept of polymorphic task hierarchies is illustrated in Fig. 55.2. Alternative task decomposition is called a "decomposition style," or simply a "style," and is to be given by designers an appropriate descriptive name. Alternative task subhierarchies are attached to their respective styles. The example polymorphic task hierarchy of Fig. 55.2 indicates the way two alternative dialogue styles for a "Delete File" task can be designed—one exhibiting direct manipulation properties with object-function syntax (e.g., the file object is selected prior to operation to be applied) with no confirmation, the other realizing modal dialogue with a function-object syntax (e.g., the delete function is selected, followed by the identification of the target file) and confirmation.

Additionally, the example demonstrates the case of physical specialization. Since selection is an abstract task, it is possible to design alternative ways for physically instantiating the selection dialogue (see Figure 55.2, lower-part), via scanning techniques for motor-impaired users, via 3-D hand pointing on 3-D auditory cues for blind people, via enclosing areas (e.g., irregular rubber banding) for sighted users, and via Braille output and keyboard input for deaf and blind users. The unified user interface design method does not require the designer to follow the polymorphic task decomposition all the way down the user-task hierarchy, until primitive actions are met. A nonpolymorphic task can be specialized at any level, following any design method chosen by the interface designer. For instance, in Fig. 55.2 (lower part) graphical mock ups are employed to describe each of the alternative physical instantiations of the abstract selection task. It should be noted that the interface designer is not constrained to using a particular model, such as CSP operators, for describing user actions for device-level interaction (e.g., drawing, drag-and-drop, concurrent input). Instead, an alternative may be preferred, such as an event-based representation, such as ERL (Hill, 1986) or UAN (Hartson, Siochi, & Hix, 1989).

As discussed in more detail in the subsequent sections, design polymorphism entails a decision-making capability for context-sensitive selection among alternative artifacts, to assemble a suitable interface instance, while task operators support temporal relationships and access restrictions applied to the interactive facilities of a particular interface instance.

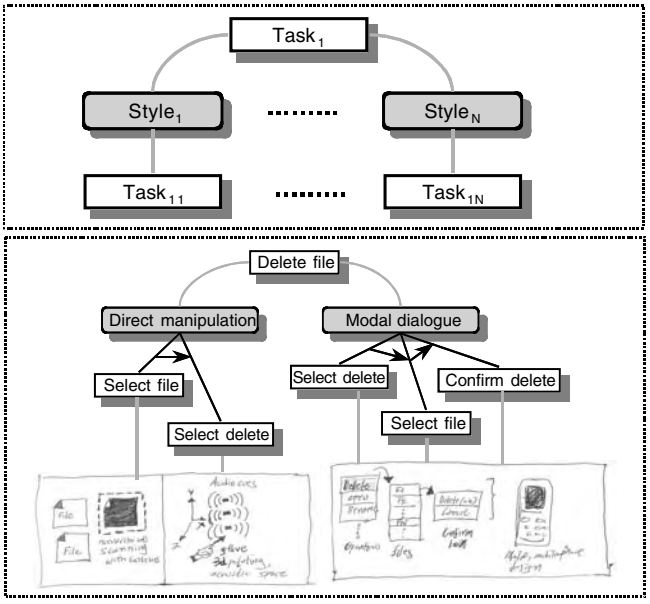


FIGURE 55.2. The polymorphic task hierarchy concept, where alternative decomposition "styles" are supported (upper part), and an exemplary polymorphic decomposition, which includes physical design annotation (lower part).

The Conduct of the Method

In the unified user interface design method, there are three categories of design artifacts, all of which are subject to polymorphism on the basis of varying user- and usage-context parameter values. These three categories are (see Fig. 55.3)

1. User tasks, relating to what the user has to do; user tasks are the center of the polymorphic task decomposition process.
2. System tasks, representing what the system has to do, or how it responds to particular user actions (e.g., feedback); in the polymorphic task decomposition process, system tasks are treated in the same manner as user tasks.
3. Physical design that concerns the various physical interface components on which user actions corresponding to the associated user task are to be performed; the physical structure may also be subject to polymorphism.

System tasks and user tasks may be freely combined within task formulas, defining how sequences of user-initiated actions and system-driven actions interrelate. The physical design, providing the interaction context, is always associated with a particular user or system task. It provides the physical dialogue pattern associated to a task-structure definition. Hence, it plays the role of annotating the task hierarchy with physical design information. An example of such annotation is shown in Fig. 55.2, where the physical designs for the "Select Delete" task are explicitly depicted.

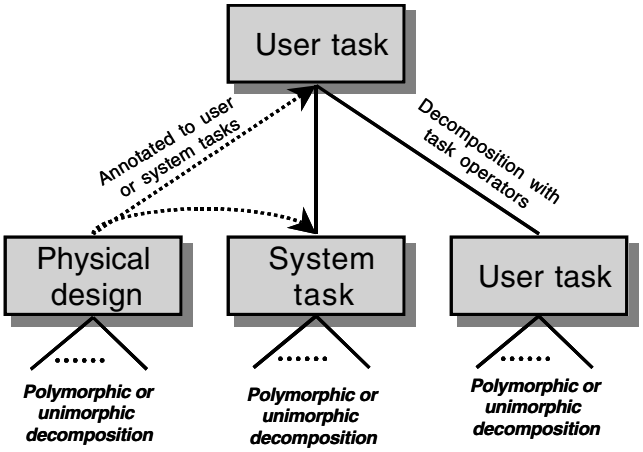


FIGURE 55.3. The three artifact categories in the Unified User Interface design method, for which polymorphism may be applied, and how they relate to each other.

In some cases, given a particular user task, there is a need for differentiated physical interaction contexts, depending on user- and usage-context parameter values. Hence, even though the task decomposition is not affected (e.g., the same user actions are to be performed), the physical design may have to be altered. One such representative example is relevant to changing particular graphical attributes, on the basis of ethnographic user attributes. For instance, Marcus (1996) discussed the choice of different iconic representations, background patterns, visual message structure, and so forth, on the basis of cultural background. User tasks, and in certain cases, system tasks, need not always be related to physical interaction, but may represent abstraction on either user or system actions. For instance, if the user has to perform a selection task, then, clearly, the physical means of performing such a task are not explicitly defined, unless the dialogue steps to perform selection are further decomposed.

This notion of continuous refinement and hierarchical analysis, starting from higher level abstract artifacts, and incrementally specializing towards the physical level of interaction, is fundamental in the context of hierarchical behavior analysis, either regarding tasks that humans have to perform (Johnson, Johnson, Waddington, & Shouls, 1988), or when it concerns functional system design (Saldarini, 1989). At the core of the unified user interface design method is the polymorphic task decomposition process, which follows the methodology of abstract task definition and incremental specialization, where tasks may be hierarchically analyzed through various alternative schemes. In such a recursive process, involving tasks ranging from the abstract task level, to specific physical actions, decomposition is applied either in a traditional unimorphic fashion or by means of alternative styles. The overall process is illustrated in Fig. 55.4; the decomposition starts from abstract or physical task design, depending on whether top-level user tasks can be defined as being abstract or not. Next follows the description of the various transitions (e.g., design specialization steps) from each of the four states illustrated in the process state diagram of Fig. 55.4.

Transitions from the abstract task design state. An abstract task can be decomposed either in a polymorphic fashion, if user- and usage-context attribute values pose the necessity for alternative dialogue patterns, or in a traditional manner, following a unimorphic decomposition scheme. In the case of a unimorphic decomposition scheme, the transition is realized via a decomposition action, leading to the task hierarchy decomposition state. In the case of a polymorphic decomposition, the transition is realized via a polymorphose action, leading to the alternative subhierarchies design state.

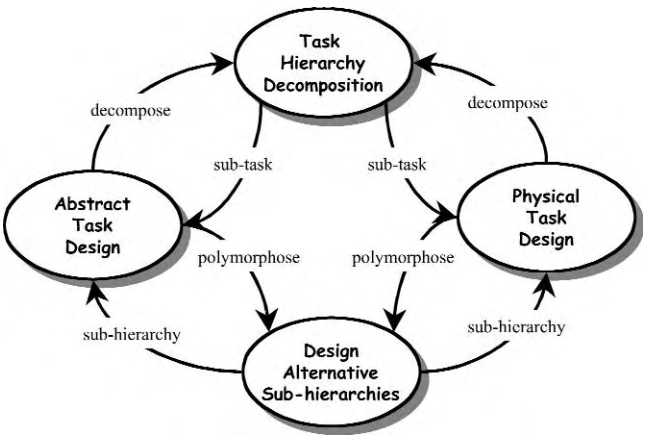


FIGURE 55.4. The polymorphic task decomposition process in the Unified User Interface design method.

Transitions from the alternative subhierarchies design state. Reaching this state means that the required alternative dialogue styles have been identified, each initiating a distinct subhierarchy decomposition process. Hence, each such subhierarchy initiates its own instance of polymorphic task decomposition process. While initiating each distinct process, the designer may start either from the abstract task design state or from the physical task design state. The former is pursued if the top-level task of the particular subhierarchy is an abstract one. In contrast, the latter option is relevant in case that the top-level task explicitly engages physical interaction issues.

Transitions from the task hierarchy decomposition state. From this state, the subtasks identified need to be further decomposed. For each subtask at the abstract level, there is a subtask transition to the abstract task design state. Otherwise, if the subtask explicitly engages physical interaction means, a subtask transition is taken to the physical task design state.

Transitions from the physical task design state. Physical tasks may be further decomposed either in a unimorphic fashion or in a polymorphic fashion. These two alternative design possibilities are indicated by decompose and polymorphose transitions respectively.

Relationships Among Alternative Styles

The need for alternative styles emerges during the design process when it is identified that there are particular user- and usage-context attribute values not addressed by the dialogue artifacts that have already been designed. Starting from this observation, one could argue, all alternative styles, for a particular polymorphic artifact, are mutually exclusive to each other (in this context, exclusion means that, at runtime, only one of those styles may be active).

However, cases in which it is meaningful to make artifacts belonging to alternative styles exist, concurrently available in a single adapted interface instance. A typical case concerns file management tasks, where two alternative but compatible artifacts may coexist during runtime, such as a direct-manipulation one and a command-based one, such as task-level multimodality. In the unified user interface design method, four design relationships between alternative styles are distinguished (see Fig. 55.5), defining whether alternative styles may be concurrently present at runtime. These four fundamental relationships reflect pragmatic, real-world design scenarios.

Exclusion. The exclusion relationship is applied when the various alternative styles are deemed usable only within the

Exclusion	*Relates many styles*. Only one from the alternative styles may be present.
Compatibility	*Relates many styles*. Any of the alternative styles may be present.
Substitution	*Relates two groups of styles together*. When the second is made "active" at run-time, the first should be "deactivated".
Augmentation	*Relates one style with a group of styles*. On the presence of any style from the group at run-time, the single style may be also "activated".

FIGURE 55.5. Design relationships among alternative styles, and their run-time interpretation.

space of their target user- and usage-context attribute values. For instance, assume that two alternative artifacts for a particular subtask are being designed, aiming to address the user-expertise attributes: one targeted to users qualified as novice, and the other targeted to expert users. Then, these two are defined to be mutually exclusive to each other, since it is probably meaningless to activate concurrently both dialogue patterns. For example, at runtime a novice user might be offered a functionally simple alternative of a task, where an expert user would be provided with additional functionality and greater freedom in selecting different ways to accomplish the same task.

Compatibility. Compatibility is useful among alternative styles for which the concurrent presence during interaction allows the user to perform certain actions in alternative ways, without introducing usability problems. The most important application of compatibility is for task multimodality, as it has been previously discussed for the file management tasks.

Substitution. Substitution has a very strong connection with adaptivity techniques. It is applied in cases where, during interaction, it is decided that some dialogue patterns need to be substituted by others. For instance, the ordering and the arrangement of certain operations may change on the basis of monitoring data collected during interaction, through which information such as frequency of use and repeated usage patterns can be extracted. Hence, particular physical design styles would need to be cancelled, while appropriate alternatives would need to be activated. This sequence of actions, for example, cancellation followed by activation, is the realization of substitution. Thus, in the general case, substitution involves two groups of styles: some styles are cancelled and substituted by other styles that are activated afterwards.

Augmentation. Augmentation aims to enhance the interaction with a particular style that is found to be valid but not sufficient to facilitate the user's task. To illustrate this point, assume that, during interaction, the user interface detects that the user is unable to perform a certain task. This would trigger an adaptation (in the form of adaptive action) aiming to provide task-sensitive guidance to the user. Such an action should not aim to invalidate the active style (by means of style substitution), but rather to augment the user's capability to accomplish the task more effectively by providing informative feedback. Such feedback can be realized through a separate but compatible style. Therefore, it follows that the augmentation relationship can be assigned to two styles when one can be used to enhance the interaction while the other is active. Thus, for instance, the adaptive prompting dialogue pattern, which provides task-oriented help, may be related via an augmentation relationship with all alternative styles (of a specific task) provided that it is compatible with them.

UNIFIED INTERFACE ENGINEERING

In the context of a unified user interface, upon startup and during runtime, the software user interface relies on the particular user and context profiles to assemble the eventual interface on the fly, collecting and gluing together the constituent interface components required for the particular end user and usage context. Such constituent components are the alternative artifacts identified during the user interface design process, which need to be appropriately transformed in the development phase to an implementation form.

Effectively, a unified user interface consists of runtime components, each with a distinctive role in performing at runtime a type of an interface assembly process, by selecting the most appropriate dialogue patterns from the available implemented design space (e.g., the organized collection of all dialogue artifacts produced during the design phase). Examples of how such intelligent selection and assembly of user interface components for adapted interaction delivery are provided in Fig. 55.6 and 55.7, taken from the AVANTI Project (Stephanidis et al., 2001).

A unified user interface does not constitute a monolithic software system but becomes a distributed architecture consisting of independent intercommunicating components, possibly implemented with different software methods/tools and residing at different physical locations. These components cooperate to perform adaptation according to the individual end-user attributes and the particular usage context. At runtime, the overall adapted interface behavior is realized by two complementary classes of system-initiated actions:

1. Adaptations driven from initial user and context information, acquired without performing interaction monitoring analysis (e.g., what is known before starting observing the user or the usage context)
2. Adaptations decided on the basis of information inferred or extracted by performing interaction-monitoring analysis (e.g., what is learned by observing the user or the usage context)

The former behavior is referred to as "adaptability" (e.g., initial automatic adaptation, performed before initiation of interaction), reflecting the capability of the interface to proactively and automatically tailor itself to the attributes of each individual end user. The latter behavior is referred to as "adaptivity" (e.g., continuous automatic adaptation) and characterizes the capability of the interface to cope with the dynamically changing/evolving characteristics of users and usage contexts. Adaptability is crucial to ensure accessibility, since it is essential to provide, before initiation of interaction, a fully accessible interface instance to each individual end user. Adaptivity can be applied only on accessible running interface instances (e.g., ones with which the user is capable of performing interaction), since in-

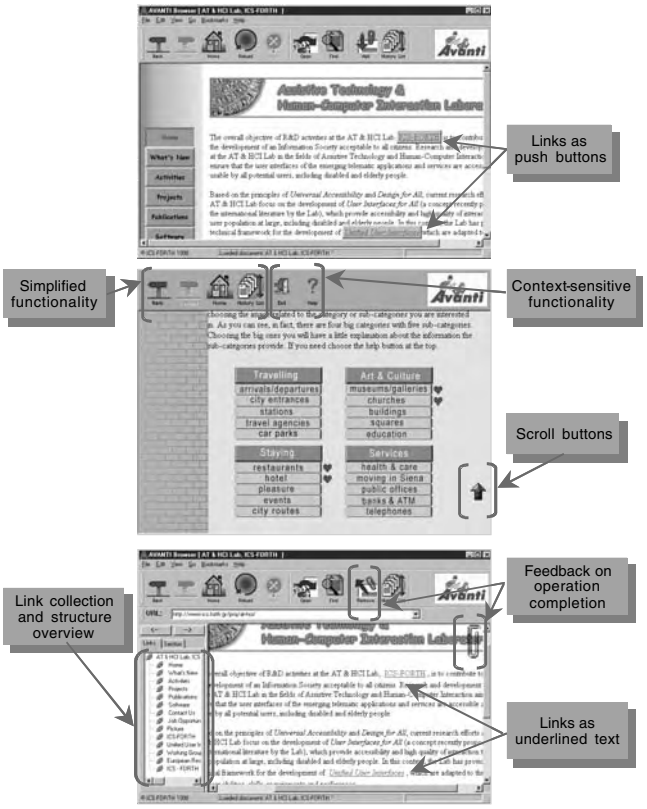


FIGURE 55.6. Three different interface versions of the AVANTI browser produced by adaptation-oriented selection of different alternative styles.

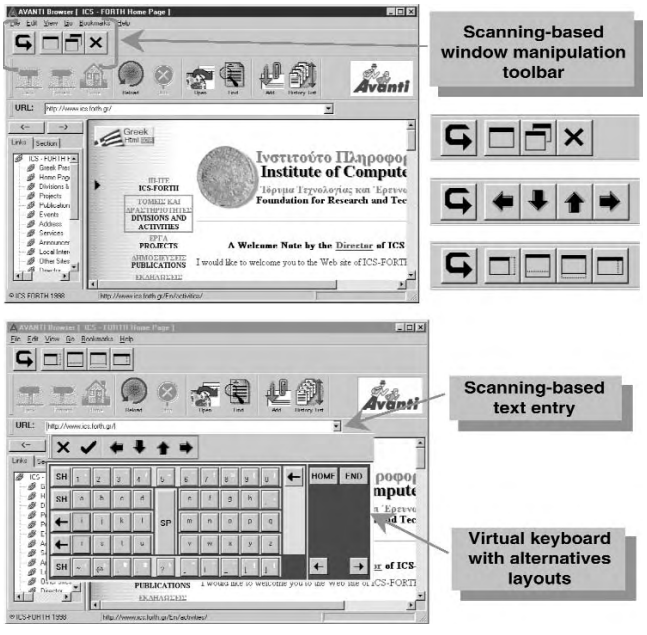


FIGURE 55.7. Alternative augmentation-oriented styles for motor-impaired user access activated at run-time in the AVANTI browser.

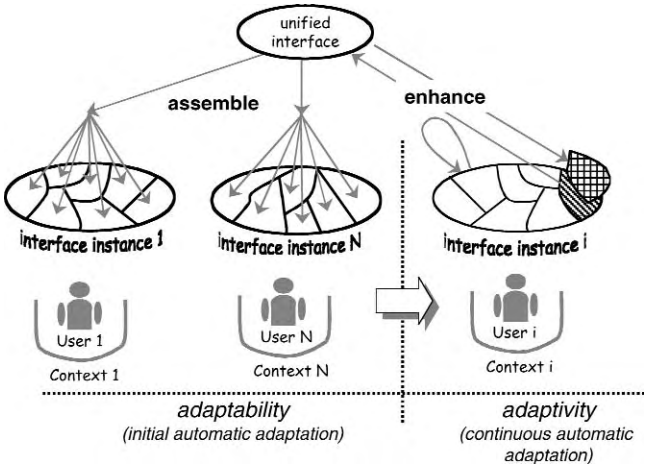


FIGURE 55.8. The complementary roles of adaptability (left) and adaptivity (right) as realised in Unified User Interfaces, to provide user- and usage-context- adapted behaviour.

teraction monitoring is required for the identification of changing or emerging decision parameters that may drive dynamic interface enhancements. The complementary roles of adaptability and adaptivity are depicted in Fig. 55.8. This fundamental distinction is made due to the different runtime control requirements between those two key classes of adaptation behaviors, requiring different software engineering policies.

Architectural Overview

In this section, the runtime architecture for unified user interfaces will be discussed, providing an outline of the adopted architectural components with information regarding (a) the functional role, (b) the runtime behavior, (c) the encapsulated context, and (d) the implementation method. The components of the unified user interface architecture are (see Fig. 55.9)

- User-Information Server (UIS)
- Context Parameters Server (CPS)
- Decision Making Component (DMC)
- Dialogue Patterns Component (DPC)

User-information server. Its functional role is to supply user attribute values (a) known offline, without performing interaction-monitoring analysis (e.g., motor/sensory abilities, age, nationality, etc.) and (b) detected online, from real-time interaction-monitoring analysis (e.g., fatigue, loss of orientation, inability to perform the task, interaction preferences, etc.). During runtime, it plays a two-fold role: (a) it constitutes a server that maintains and provides information regarding individual user profiles, and (b) it encompasses user-representation schemes, knowledge processing components, and design information dynamically to detect user properties or characteristics. This component may need to employ alternative ways of representing user-oriented information. In this sense, a repository of user profiles serves as a central database of individual

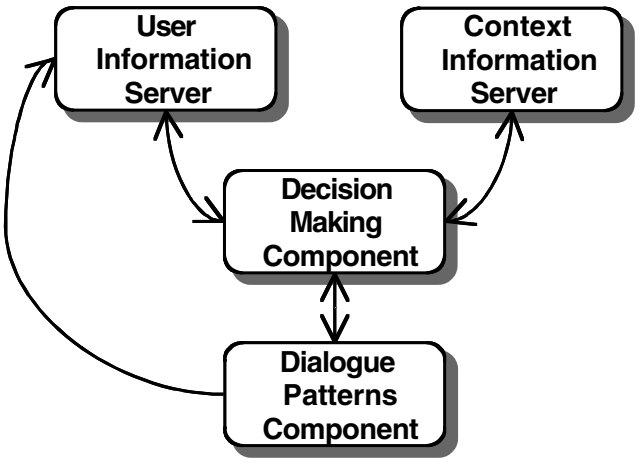


FIGURE 55.9. The four basic components of the Unified User Interface macro-architecture outlining run-time communication links.

user information (e.g., registry). In many cases, a profile structure as a typical list of typed attributes will suffice shown; this model, though quite simple, is proved in real practice to be very powerful and flexible (can be stored in a database, thus turning the profile manager to a remotely accessed database). Additionally, more sophisticated user representation and modeling methods can be employed, including support for stereotypes of particular user categories. In case dynamic user-attribute detection is to be supported, the content may include dynamically collected interaction monitoring information, design information, and knowledge-processing components.

From a knowledge representation point of view, static or preexisting user knowledge may be encoded in any appropriate form, depending on the type of information, the user information server should feed to the decision making process. Moreover, additional knowledge-based components may be employed for processing retrieved user profiles, drawing assumptions about the user, or updating the original user profiles. In Fig. 55.10, the internal architecture of the user-information server employed in the AVANTI web browser is presented. It should be noted that the first version of the AVANTI web browser produced in the context of the AVANTI Project employed BGP-MS (Kobsa & Pohl, 1995) for the role of the user-information server. The profile manager has been implemented as a database of profiles. The two other subsystems (e.g., monitoring manager, modeling, and inference) are needed only in case dynamic user-attribute detection is required. The interaction monitoring history has been implemented as a time-stamped list of monitoring events (the structure of monitoring events is described in the analysis of communication semantics) annotated with simple dialogue design context information (e.g., just the subtask name). In the user models, all of the types of dynamically detected user attributes have been identified (e.g., inability to perform a task, loss of orientation—those were actually the two dynamically detectable attributes required by the design in the AVANTI web browser). Each such attribute is associated with its corresponding behavioral action patterns. In the specific case, the representation of the behavioral patterns has been implemented together with the pattern-matching component, by means of state automata.

For instance, one heuristic pattern to detect loss of orientation has been defined as follows: the user moves the cursor inside the web-page display area, without selecting a link, for more than *N* seconds. The state automaton (see Fig. 55.11) starts recording mouse moves in the page area, based on incoming monitored mouse moves, while finally triggering detection when no intermediate activity is successfully performed by the user. This worked fine from an implementation point of view. However, from the user interface design point of view, all such heuristic assumptions and behavioral patterns had to be extensively verified with real users so as to assert the relationship between the observable user behavior and the particular inferred user attributes. This is a common issue in all adaptive

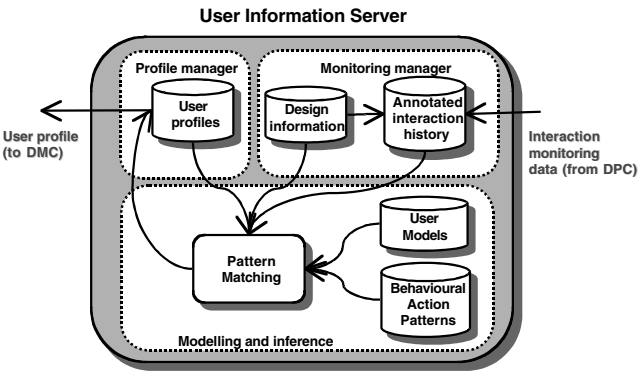


FIGURE 55.10. Internal micro-architecture of the User Information Server. User profile is posted to the Decision Making Component (DMC), while interaction-monitoring information is received form the Dialogue Patterns Component (DPC).

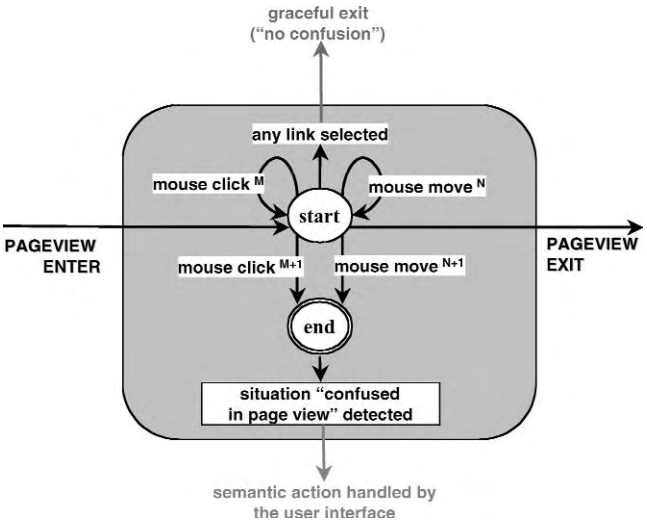


FIGURE 55.11. An example of an augmented state automaton used as an implementation technique for behavioural patterns; the state automaton is directly hard-coded in the UIS side, while multiple such automata co-exist.

systems that employ heuristics for detecting user attributes at runtime, practically meaning that the validity of the assumptions inferred is dependent on the appropriateness of the specific user-action patterns chosen.

Context parameters server. The purpose of this component is to supply context attribute values (machine and environment) of two types: (a) (potentially) invariant, meaning unlikely to change during interaction, such as peripheral equipment, and (b) variant, dynamically changing during interaction (e.g., environment noise, failure of particular equipment, etc.). This component is not intended to support device independence, but to provide device awareness. Its purpose is to enable the decision-making component to select those interaction patterns, which, apart from fitting the particular end-user attributes, are also appropriate for the type of equipment available on the end-user machine.

The usage-context attribute values are communicated to the decision-making component before the initiation of interaction. Additionally, during interaction, some dynamically changing usage-context parameters may also be fed to the decision-making component for decisions regarding adaptive behavior. For instance, assume that the initial decision for selecting feedback leads to the use of audio effects. Then, the dynamic detection of an increase in environmental noise may result in a runtime decision to switch to visual feedback (the underlying assumption being that such a decision does not conflict with other constraints).

This component encompasses a listing of the various invariant properties and equipment of the target machine (e.g., handheld binary switches, speech synthesizer for English, high-resolution display, mode 16bits, 1024×768, noisy environment, etc.). In this context, the more information regarding the characteristics of the target environment and machine is encapsulated, especially concerning I/O devices, the better adaptation can be achieved (information initially appearing redundant is likely to be used in future adaptation-oriented extensions).

The registry of environment properties and available equipment can be implemented easily as a profile manager in the form of a database. Such information will be communicated to the decision-making component as attribute/value pairs. However, if usage-context information is to be dynamically collected, such as environment noise, or reduction of network bandwidth, the installation of proper hardware sensors or software monitors becomes mandatory.

Decision-making component. The role of this component is to decide, at runtime, the necessary adaptability and adaptivity actions and subsequently to communicate those to the dialogue patterns component (the latter being responsible for applying adaptation-oriented decisions). To decide adaptation, this component performs a kind of rule-based knowledge processing, to match end-user and usage-context attribute values to the corresponding dialogue artifacts, for all the various dialogue contexts.

This module encompasses the logic for deciding the necessary adaptation actions, on the basis of the user- and context-attribute values, received from the user-information server and the context parameters server, respectively. Such attribute values will be supplied to the decision-making component, prior to the initiation of interaction within different dialogue contexts (e.g., initial values, resulting in initial interface adaptation), as well as during interaction (e.g., changes in particular values, or detection of new values, resulting in dynamic interface adaptations).

In the proposed approach, the encapsulated adaptation logic should reflect predefined decisions during the design stage. In other words, the inference mechanisms employ well-defined decision patterns that have been validated during the design phase of the various alternative dialogue artifacts. In practice, this approach leads to a rule-based implementation, in which embedded knowledge reflects adaptation rules that have been already constructed and documented as part of the design stage. This decision-making policy is motivated by the assumption that if a human designer cannot decide upon adaptation for a dialogue context, given a particular end user and usage context, then a valid adaptation decision cannot be taken by a knowledge-based system at runtime. Later in this paper, while discussing some implementation details of the AVANTI web browser, specific excerpts from the rule base of the decision engine will be discussed.

The first remark regarding the implementation of decision making concerns the apparent awareness regarding (a) the various alternative dialogue artifacts (how they are named—e.g., virtual keyboard, for which dialogue context they have been designed—e.g., *http address text field)*, and (b) user- and usage-context attribute names, and their respective value domains (e.g., attribute age being integer in range 5 . . . 110).

The second issue concerns the input to the decision process, being individual user- and usage-context attribute values. Those are received at runtime both from the user-information server and from the context information server, either by request (e.g., the decision-making component takes the initiative to request the end-user and usage-context profile at start up to draw adaptability decisions) or by notification (e.g., when the user-information server draws assumptions regarding dynamic user attributes, or when the context parameters server identifies dynamic context attributes).

The third issue concerns the format and structure of knowledge representation. In all developments that we have carried out, it has been proved that a rule based logic implementation is practically adequate. Moreover, all interface designers engaged in the design process emphasized that this type of knowledge representation approach is far more close to their own way of rule-based thinking in deciding adaptation. This remark has led to excluding, at a very early stage, other possible approaches, such as heuristic pattern matching, weighting factor matrices, or probabilistic decision networks.

The final issue concerned the representation of the outcomes of the decision process in a form suitable for being communicated and easily interpreted by the dialogue patterns component. In this context, it has been practically proved that two categories of dialogue control actions suffice to communicate adaptation decisions: (a) activation of specific dialogue components, and (b) cancellation of previously activated dialogue components. These two categories of adaptation actions provide the expressive power necessary for communicating the dialogue-component manipulation requirements that realize both adaptability and adaptivity. Substitution is modeled by a message

containing a series of cancellation actions (e.g., the dialogue components to be substituted), followed by the necessary number of activation actions (e.g., which dialogue components to activate in place of the cancelled components). Therefore, the transmission of those commands in a single message (e.g., cancellation actions followed by activation actions) is to be used for implementing a substitution action. The need to send in one message packaged information regarding the cancelled component, together with the components which take its place, emerges when the implemented interface requires knowledge of all (or some) of the newly created components during interaction. For instance, if the new components include a container (e.g., a window object) with various embedded objects, and if upon the creation of the container information on the number and type of the particular contained objects is needed, it is necessary to ensure that all the relevant information (e.g., all engaged components) is received as a single message. It should be noted that, since each activation/cancellation command always carries its target UI component identification, it is possible to engage in substitution requests components that are not necessarily part of the same physical dialogue artifact. In addition, the decision to apply substitution is the responsibility of the decision-making component.

One issue regarding the expressive power of activation and cancellation decisions categories concerns the way dynamic interface updates (e.g., changing style or appearance, without closing or opening interface objects) can be effectively addressed. The answer to this question is related to the specific connotation attributed to the notion of a dialogue component. A dialogue component may not only implement physical dialogue context, such as a window and embedded objects, but may concern the activation of dialogue control policies or be realized as a particular sequence of interface manipulation actions. In this sense, the interface updates are to be collected in an appropriate dialogue implementation component (e.g., a program function, an object class, a library module) to be subsequently activated (e.g., called) when a corresponding activation message is received. This is the specific approach taken in the AVANTI web browser, which, from a software engineering point of view, enabled a better organization of the implementation modules around common design roles.

Dialogue patterns component. This component is responsible for supplying the software implementation of all the dialogue artifacts that have been identified in the design process. Such implemented components may vary from dialogue artifacts that are common across different user- and usage-context attribute values (e.g., no adaptation needed), to dialogue artifacts that will map to individual attribute values (e.g., alternative designs have been necessitated for adapted interaction). Additionally, as it has been previously mentioned, apart from implementing physical context, various components may implement dialogue-sequencing control, perform interface manipulation actions, maintain shared dialogue state logic, or apply interaction monitoring.

The dialogue patterns component should be capable of applying at runtime, activation or cancellation decisions originated from the decision-making component. Additionally, interaction-monitoring components may need to be dynamically installed/uninstalled on particular physical dialogue components. This behavior will serve the runtime interaction monitoring control requests from the user-information server, to provide continuous interaction monitoring information back to the user-information server for further intelligent processing. The dialogue patterns component either embeds the software implementation of the various dialogue components or is aware of where those components physically reside by employing dynamic query, retrieval, and activation methods. The former is the typical method that can be used if the software implementation of the components is provided locally by means of software modules, libraries, or resident installed components. Usually, most of the implementation is to be carried out in a single programming language. The latter approach reflects the scenario in which distinct components are implemented on top of component-ware technologies, usually residing in local/remote component repositories (also called "registries" or "directories"), enabling reuse with dynamic deployment.

In the development of the AVANTI web browser, a combination of these two approaches has been employed by implementing most of the common dialogue components into a single language (actually in C++ by employing all the necessary toolkit libraries) while implementing some of the alternative dialogue artifacts as independent Active X components that were located and employed on the fly. The experience from the software development of the AVANTI web browser has proved that (a) the single language paradigm makes it far easier to perform quick implementation and testing of interface components, and (b) the component-based approach largely promotes binary format reuse of implemented dialogue components while offering far better support for dynamic interface assembly, which is the central engineering concept of unified user interfaces (this issue we will be elaborated upon in the conclusion section of the paper).

The microarchitecture of the dialogue patterns component internally employed in the AVANTI web browser, as outlined in Fig. 55.12, emphasizes internal organization to enable extensibility and evolution by adding new dialogue components. Additionally, it reflects the key role of the dialogue patterns component in applying adaptation decisions. The internal components are as follows.

The activation dispatcher locates the source of implementation of a component (or simply uses its API, if it is a locally used

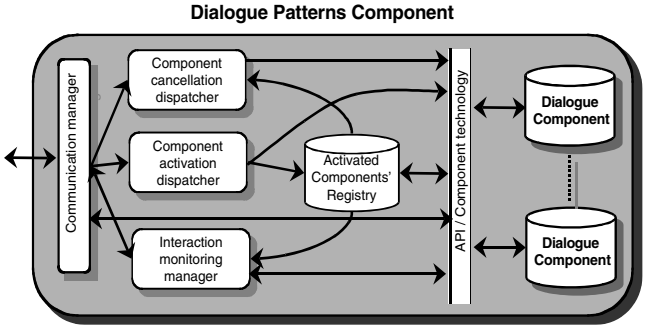


FIGURE 55.12. Internal micro-architecture of the Dialogue Patterns Component.

library) to activate it. In this sense, activation may imply a typical instantiation in OOP terms, calling of particular service functions, or activating a remotely located object. After a component is activated, if cancellation is to be applied to this component, it is further registered in a local registry of activated components. In this registry, the indexing parameters used are the particular dialogue context (e.g., subtask, for instance "http address field") and the artifact design descriptor (e.g., unique descriptive name provided during the design phase—for instance, "virtual keyboard"). For some categories of components, cancellation may not be defined during the design process, meaning there is no reason to register those at runtime for possible future cancellation (e.g., components with a temporal nature that perform only some interface update activities).

The cancellation dispatcher locates a component based on its indexing parameters and calls for cancellation. This may imply a typical destruction in OOP terms, calling internally particular service functions that may typically perform the unobtrusive removal of the physical view of the cancelled component or the release of a remote object instance. After cancellation is performed, the component instance is removed from the local registry.

The monitoring manager plays a two-fold role: (a) It applies monitoring control requests originated from the user-information server by first locating the corresponding dialogue components and then requesting the installation (or uninstallation) of the particular monitoring policy (this requires implementation additions in dialogue components, for performing interaction monitoring and for activating or deactivating the interaction monitoring behavior); and (b) it receives interaction monitoring notifications from dialogue components and posts those to the user-information server.

The communication manager is responsible for dispatching incoming communication (activation, cancellation, and monitoring control) and posting outgoing communication (monitoring data and initial adaptation requests). One might observe an explicit link between the dialogue components and the communication manager. This reflects the initiation of interaction in which the dialogue control logic (residing within dialogue components) requests iteratively the application of decision making (from the decision-making component). Such requests will need to be posted for all cases involving dialogue component alternatives for which adapted selection has to be appropriately performed.

The dialogue components typically encompass the real implementation of physical dialogues, dialogue control logic, and interaction monitoring method. In practice, it is hard to accomplish isolated implementation of the dialogue artifacts as independent black boxes that can be combined and assembled on the fly by independent controlling software. In most designs, it is common that physical dialogue artifacts are contained inside other physical artifacts. In this case, if there are alternative versions of the embedded artifacts, it turns out that to make containers fully orthogonal and independent with respect to the contained, one has to support intensive parameterization and pay a heavier implementation overhead. However, the gains are that the implementation of contained artifacts can be independently reused across different applications, while in the more monolithic approach, reuse requires deployment of the container code (and recursively, of its container too, if it is contained as well).

Adaptability and Adaptivity Cycles

The completion of an adaptation cycle, being either adaptability or adaptivity, is realized in a number of distributed processing stages performed by the various components of the unified architecture. During these stages, the components communicate with each other, requesting or delivering specific pieces of information. Fig. 55.13 outlines the processing steps for performing both the initial adaptability cycle (to be executed only once), as well as the two types of adaptivity cycles (e.g., one starting from dynamic context attribute values and another starting from interaction monitoring control). Local actions indicated within components (in each of the four columns) are either outgoing messages, shown in bold typeface, or necessary internal processing, illustrated via shaded rectangles.

Dynamic User Interface Assembly

The concept of dynamic interface assembly reflects the key runtime mechanisms to support adaptability in unified user interfaces. Previous work in adaptive interaction involved techniques such as detection of user attributes, adaptive prompting, and localized lexical-level modifications (e.g., rearranging menu options, or adding/removing operation buttons). The issue of making the interface fit from the beginning to individual users has been addressed in the past mainly as a configuration problem, requiring interface developers to supply configuration editors so that end users could fit the interface to their particular preferences. However, such methods are limited to fine tuning some lexical-level aspects of the interface (e.g., toolbars, menus), while they always require explicit user intervention, for example, there is no automation. In this context, the notion of adaptability, as realized in unified user interfaces, offers new possibilities for automatically adapted interactions, while the architecture and runtime mechanisms to accomplish dynamic interface assembly constitute a unique software engineering perspective.

Some similarities with dynamic interface assembly can be found in typical web-based applications delivering dynamic content. The software engineering methods employed in such cases are based on the construction of application templates (technologies such as Active Server Pages, ASP, by Microsoft or Java Server Pages, JSP, by JavaSoft are usually employed) with embedded queries for dynamic information retrieval, delivering to the user a web page assembled on the fly. In this case, there are no alternative embedded components—just content to be dynamically retrieved—while the web-page assembly technique is mandatory when HTML-based web pages are to be delivered to the end user (in HTML, each time the content changes, a different HTML page has to be written). However, in case a full-fledged embedded component is developed (e.g., ActiveX object or Java Applet), no runtime assembly is required, since the embedded application internally manages content extraction and display, as a common desktop information retrieval application.

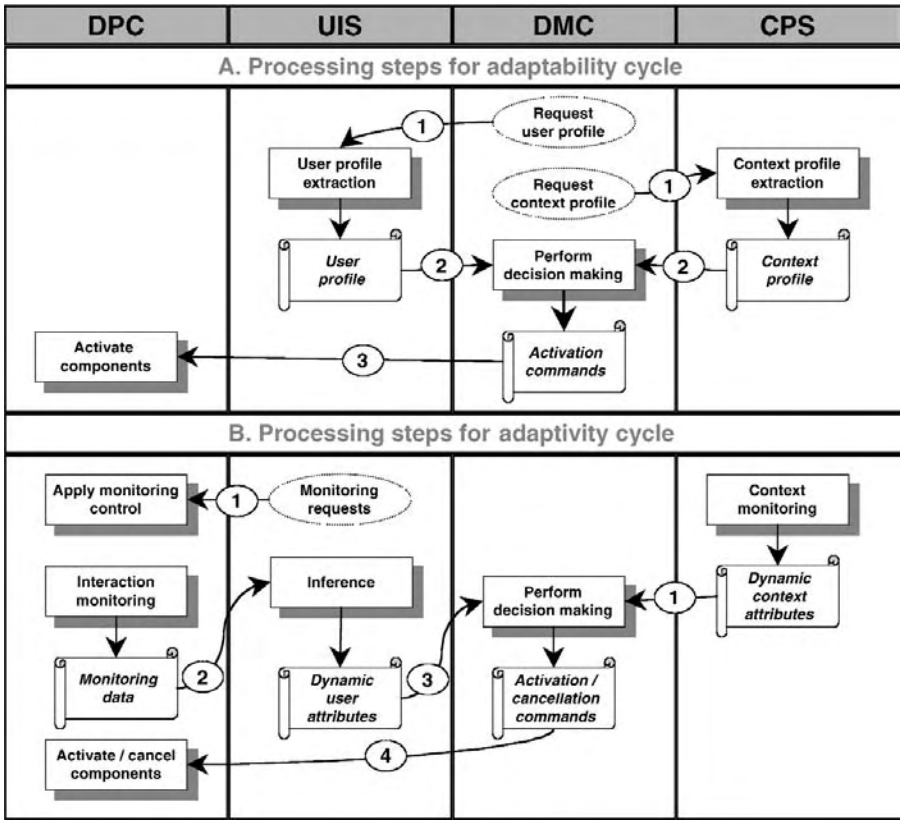


FIGURE 55.13. Processing steps, engaging communication among architectural components, to perform initial adaptability cycles (A), as well as the two types of adaptivity cycles (B). Requests originated from Dialogue Patterns Component are indicated with dashed ovals, communicated messages are shown with special banners, processing points are drawn with shaded rectangles, while logical ordering is designated with numeric oval labels.

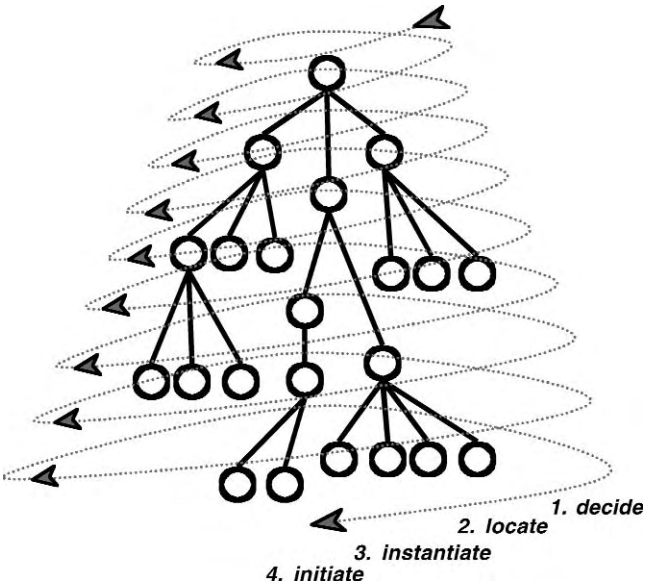


FIGURE 55.14. Illustration of the dynamic interface assembly process as an incremental hierarchical construction procedure.

The implementation of unified user interfaces is organized in hierarchically structured software templates, in which the key placeholders are parameterized container components. This hierarchical organization, as it has been reflected in the development excerpts, mirrors the fundamentally hierarchical constructional nature of interfaces. The ability to diversify and support alternatives in this hierarchy is due to containment parameterization, while the adapted assembly process is realized by selective activation, engaging remote decision making on the basis of end-user and usage-context information. The dynamic interface assembly process reflects the hierarchical traversal in the task hierarchy, starting from the root, to decide, locate, instantiate, and initiate appropriately every target user interface component (see Fig. 55.15).

This process primarily concerns the interface components that implement alternative styles. From the implementation point of view, the following software design decisions have been made:

- The task hierarchy has been implemented as a tree data structure with polymorphic nodes triggering decision-making sessions (see Fig. 55.15).

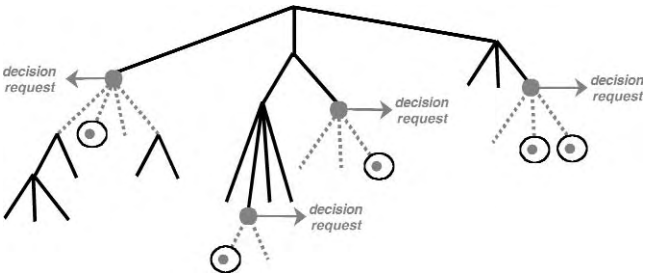


FIGURE 55.15. Illustrating the hierarchical posting of decision requests, causing decision sessions for each polymorphic task (shown with decomposition alternatives as dashed lines), and marking of selected alternative styles (i.e., interface components), after each decision session completes.

- Interface components have been implemented as distinct independent software modules, implementing generic containment application programming interfaces (APIs), while exposing a singleton control API for dynamic instantiation and name-based lookup.
- The interface assembly procedure is actually carried out via two successive hierarchical passes:
  - Execution of decision sessions, to identify the specific styles for polymorphic task contexts, which will be part of the eventually delivered user interface
  - Interface construction, through instantiation and initiation of all interface components for the decided styles

Polymorphic Containment Hierarchies

In Fig. 55.16, the concept of parametric container hierarchies is illustrated, while in Fig. 55.17, an instantiation of the concept is shown for the AVANTI web browser. Container classes expose their containment capabilities and the type of supported contained objects by defining abstract interfaces (e.g., abstract OOP classes) for all of the contained component classes.

These interfaces, defined by container class developers, constitute the programming contract between the container and the contained classes. In this manner, alternative derived contained-component classes may be instantiated at runtime as constituent elements of a container. Following the definition of polymorphic factor PL, which provides a practical metric of the number of possible alternative runtime configurations of a component (not all of which may be semantically viable), the PL of the top-level application component gives the number of the possible alternative dynamically assembled interface instances. From a programming point of view, in the AVANTI web browser, the activation control of dialogue components for runtime assembly has been mainly realized through typical library function calls. Such function calls engage object instances corresponding to dialogue components, without employing any componentware technology. Hence, this runtime assembly behavior has been accomplished without the need of locating, fetching, and combining components together. Nevertheless, efforts have been devoted to applying and testing the latter approach in real

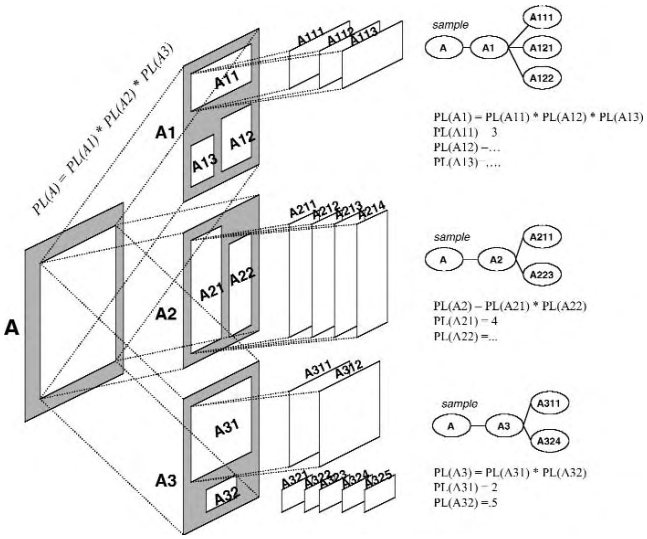


FIGURE 55.16. The structure of polymorphic containment hierarchies. Alternative contained components implement a common abstract interface that containers use, and vice versa. PL indicates the maximum polymorphism factor, which provides the total number of all possible different run-time incarnations of an interface component, recursively defined as the product of the polymorphic factors of constituent component classes. Not all PL combinations may be semantically viable.

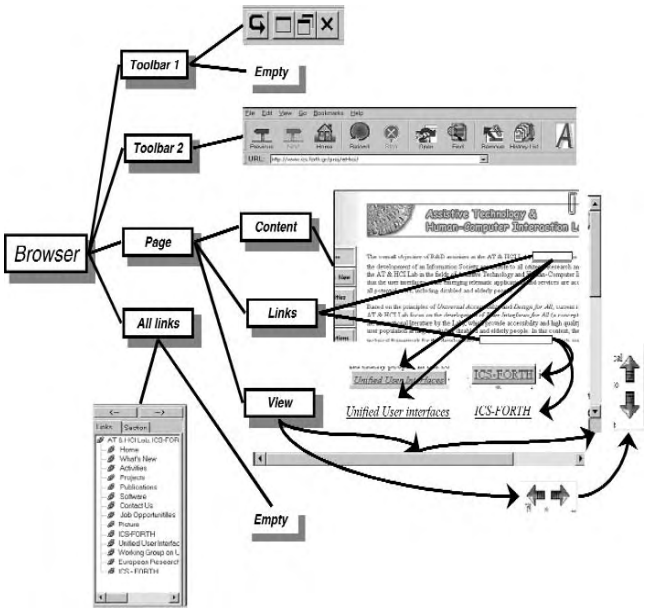


FIGURE 55.17. Parametric polymorphic containment with variant constituent components in the AVANTI browser. The indication "Empty" indicates components whose presence may have to be omitted upon dynamic interface delivery for certain user categories.

practice, by employing a component-ware technology (DCOM/ActiveX) for a limited number of dialogue components. This required a more labor-intensive implementation approach (from a C++ point of view, while isolated testing of components with VisualBasic was far easier), for packaging dialogues to make them component enabled, as well as for further activating and using them at runtime. However, there are some evident advantages:

- Dialogue components need not be carried altogether, but can be dynamically loaded, thus promoting a thin dialogue patterns component implementation.
- In effect, the core logic of the dialogue patterns component, apart from dialogue components, can be also packaged as a component itself, making it reusable across different applications.
- Automatic updates and extensions of components are directly supported, enabling new versions, or even new dialogue components (addressing more user- and usage-context attribute values), to be centrally installed in appropriate component repositories.

Decision-Making Specification

The decision-making logic is defined in independent decision blocks, each uniquely associated to a particular task context; at most one block per distinct task context may be supplied. The decision-making process is performed in independent sequential decision sessions, and each session is initiated by a request of the interface assembly module for execution of a particular initial decision block. In such a decision session, the evaluation of an arbitrary decision block may be performed, while the session completes once the computation exits from the initial decision block. The primary decision parameters are the end-user and usage-context profiles, defined as two built-in objects, such as user and context, whose attributes are syntactically accessible in the form of named attributes. The binding of attribute names to attribute values is always performed at runtime. The encapsulation of composite attributes in user and context profiles is easily allowed due to the syntactic flexibility of attributes reference. For instance, "user-abilities.vision" and "user-abilities.hearing" are syntactic sugar for "user-abilities.vision" and "user-abilities.hearing", where "abilities.vision" and "abilities.hearing" are two distinct independent ordinal attributes of the user built-in object. Consequently, even though all attributes in the DMSL language are semantically scalar, the flexibility of attribute names allows syntactical simulation of aggregate structures.

In Fig. 55.18, an example decision block is shown, being an excerpt of the implementation of the decision logic AVANTI web browser, for selecting the best alternative interface components for the link task context. The interface design relating to this adaptation decision logic is provided in Fig. 55.1.

In the DMSL language, the fundamental design relationships among alternative styles are not injected as a part of the semantics but, alternatively, concrete rule patterns are delivered, effectively mapping those relationships to implementation skeletons of decision blocks. This gives adaptation designers the freedom not to necessarily adopt those particular design relationships, in case, for instance, they do not choose to employ

```
taskcontext link [
        evaluate linktargeting;
        evaluate linkselection;
        evaluate loadconfirmation;
]
taskcontext linktargeting [
        if (user.abilities.pointing == accurate) then
                activate "manual pointing";
        else
                activate "gravity pointing";
]
taskcontext linkselection [
        if (user.webknowledge in {good, normal}) then
                activate "underlined text";
        else
                activate "push button";
]
taskcontext loadconfirmation [
        if (user.webknowledge in {low, none} or context.net==low) then
                activate "confirm dialogue";
        else
                activate "empty";
]
```

FIGURE 55.18. An example of a simple decision block to select the most appropriate delivery of web links for the individual end-user; notice that names in italics are not language keywords but are treated as string constants, i.e., user.webknowledge is syntactic sugar for user."webknowledge".

Exclusion(S_1, S_2)	Compatibility(S_1, S_2)
if (S_1.*cond*) then activate S_1; else if (S_2. *cond*) then activate S_2;	if (S_1.*cond*) then activate S_1; if (S_2. *cond*) then activate S_2;
Substitution(S_1 by S_2)	**Augmentation(S_1 by S_2)**
if (S_2. *cond and* isactive(S_1)) then [cancel S_1; activate S_2;] else if (S_1.*cond*) activate S_1;	if (S_1.*cond*) if (not isactive(S_1)) then activate S_1; else if (S_2. *cond*) then activate S_2;

FIGURE 55.19. The decision rule patterns associated to the relationships among alternative styles; the style condition is the boolean expression engaging the user and context attribute values for which the style is designed.

unified design as the adaptation-design approach. In Fig. 55.19, the DMSL decision-rule patterns are provided, for the previously described four categories of style relationships.

UNIFIED DEVELOPMENT IN AMBIENT INTELLIGENCE

Ambient intelligence is an emerging field for interactive applications and services, in which the capability to accomplish increased quality of the delivered interactive experience heavily depends on the ability to cope, in a cost effective and robust way, with the inherent software user interface implementation complexity. In this section, research results in implementing ambient interactions are reported; in particular, we discuss the way the unified user interface development has been practiced, putting forward the development challenges we had to address to accomplished user interface unification.

Ambient Interactions

One key dimension of user mobility, in a disappearing computer context, concerns the provision of high-quality, unobtrusive, and fault-tolerant dialogues for typical computer-based applications. In this context, we are focusing on a scenario according to which the basic computing machine is a pocketsize portable processing unit (PPU), without interaction-related Input/Output (I/O) hardware, where all interaction takes place dynamically with wireless devices, which can be either wearable (optional) or part of the surrounding environment infrastructure.

Following this scenario, the user carries a very small processing unit, such as the size of a credit card, with an embedded operating system (e.g., Linux) and wireless short-range radio networking (e.g., BlueTooth™). Additionally, the user may optionally collect any number of wearable wireless gadgets. Once the on button of the processing unit is pressed, the system boots, and then seeks for in-range devices capable of hosting interaction. When such devices are detected, they are appropriately employed to support interaction. At some point, some devices get out of range (e.g., they are lost), and the system tries to use the rest of available devices to maintain interaction; if the available devices do not suffice, dialogue is considered as stalled.

When new devices become in-range (e.g., they are discovered), the system tries to engage those devices in interaction, either to resume dialogue from a stalled state, or to further optimize it by offering a better interaction alternative.

This notion of mobile interaction is illustrated in Fig. 55.20, which depicts two layers of dynamically engaged interaction-capable devices: (a) Inner layer wearable devices, which the user may or may not carry, depending on the situation. Wearable devices are not anticipated to vary on the move as frequently as environment devices. And (b) Outer layer ambient/environment devices, such as the particular set of devices falling inside a wireless communication range with the user's portable processing unit. Those devices will normally vary according to the user's location in space.

The primary motivation for the reported work is based on the vision that the future computing platforms will not constitute monolithic all power in one devices, but will likely support open interconnectivity enabling users to combine the facilities offered by distinct devices on the fly. Physically distributed devices may be either wearable or available within the ambient infrastructure (either stationary or mobile) and may be connected via a wireless communication link for easier deployment. Operationally, each such device will play a specific role by exposing different processing capabilities or functions, such as character display, pointing, graphics, audio playback, speech synthesis, storage, network access, and so forth. From the hardware point of view, such devices may be wristwatches, earphones, public displays, home appliances, office equipment, car electronics, sunglasses, ATMs, and so forth.

Unified Ambient User Interfaces

The need for a unified user interface emerges when diverse target use groups need to be accommodated, with diverse requirements necessitating differentiated user interface delivery, either macroscopically, for example, the user interface is radically different, or microscopically, for example, there are specific targeted task variations implying alternative user interface components. In this context, we have concentrated in unified ambient interactions, driven by our ultimate objective to pursue ambient intelligence applications that would potentially address the needs of every individual end user.

Such a target poses severe technical challenges accessibility; however, with appropriate technical propositions, it may open new opportunities for inclusive mobile interactions and accessible ambient intelligence. The implementation details of the basic system to allow dynamic deployment of ambient devices

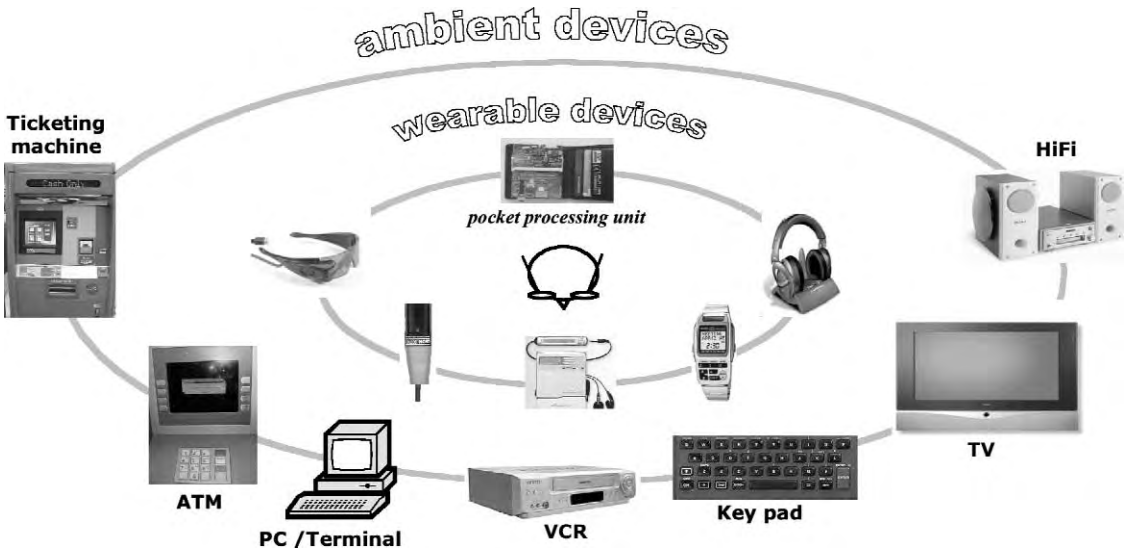


FIGURE 55.20. The two layers of dynamically engaged ambient computing resources in mobile interactions.

for interaction purposes is reported by Savidis and Stephanidis (2005), while various runtime snapshots of real-life applications and related technical reports may be found on the 2WEAR Project website (http://2wear.ics.forth.gr). Following Fig. 55.21, the system offers the following interactive facilities to the user:

- Basic mobile interface consists of:
  1. The PPU, such as the portable computer having no interaction-specific equipment, developed by MASC Ltd in the context of the 2WEAR Project (Majoe, 2003).1.
  2. The wearable wireless Bluetooth binary switches, which had to be simulated in software running in two different PDAs (still using Bluetooth).2.
  3. A wearable intelligent wristwatch device controlled and deployed by the PPU for interaction purposes, also developed by MASC Ltd in the context of the 2WEAR Project (Majoe, 2003).3.
  4. A wheel chair attached wireless Bluetooth output terminal, simulated entirely in software over a PDA.
- Ambient interface extension concerns all types of remote ambient devices, controllable over Bluetooth through the 2WEAR-specific control protocol (Savidis & Stephanidis, 2005), offering output interaction facilities such as graphics terminals, character-based terminals, audio equipment, and speech synthesizers.
- Cooperative interface extension concerns the dynamic utilization of typical desktop machines available on the ambient space that host GUI software libraries. In order for the developed system to be multiplatform, it has been decided to consider computing devices that encompass the Java Full or MicroEdition. Such computing devices are employed dynamically by the runtime interface system (on the PPU) to replicate the basic mobile interface (and its particular ambient extension) on a proximate desktop machine via a graphical interface. This interface mirroring is activated on demand by the motor-impaired user in order to enable mobile cooperation with an able user.
- The software architecture for ambient interactions (see Fig. 55.22) reflects a novel way of treating environment devices in an ambient intelligence framework: potentially every environment device exports user interface functionality (e.g., text display, audio output, graphic display, input buttons, etc.) in the form of services, called user interface microservices over Bluetooth. The proximity-based discovery mechanism of Bluetooth allows only a small cluster of devices to be made reachable from the user PPU, while the set of devices deployed for interaction may vary with time as the user moves within the environment. Some representative microservices that have been implemented in the context of the Voyager system (Savidis & Stephanidis, 2005) are:
  - Button, input control
  - Keyboard, input control
  - Numeric keypad, input control
  - Text area (character display), output control
  - Text line (character display), output control
  - Graphics display, output control
  - Menu (composite device service), input/output control
  - Text editor (composite device service), input/output control

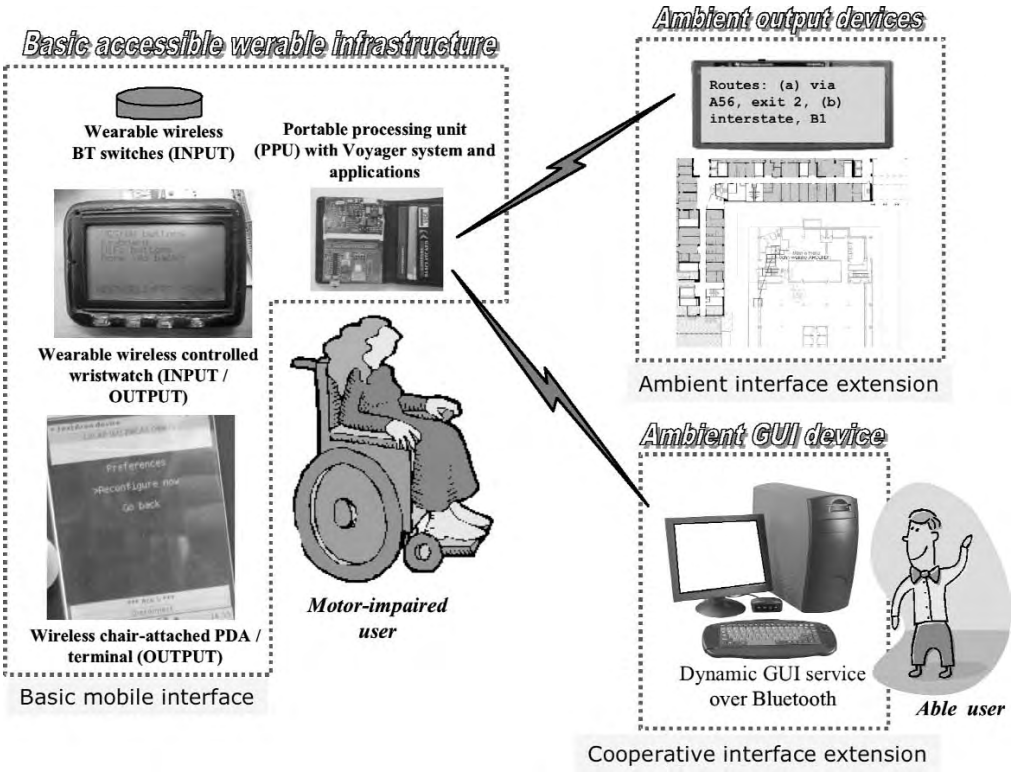


FIGURE 55.21. The system infrastructure for motor-impaired accessible and cooperative ambient dialogues.

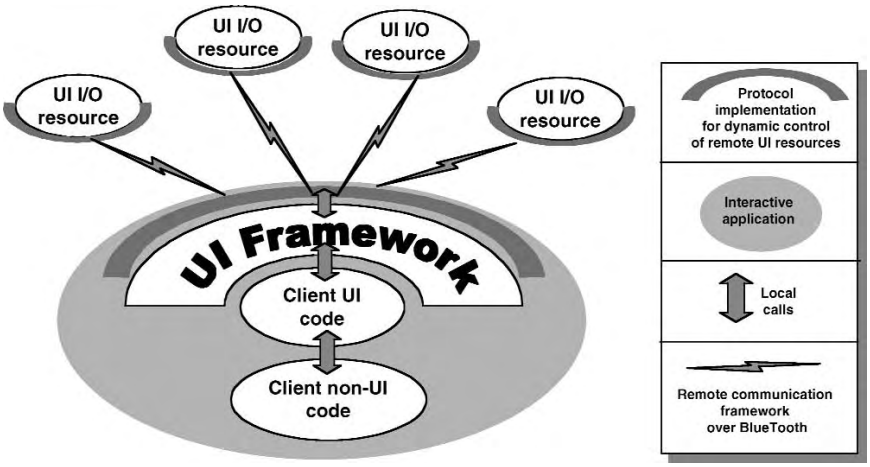


FIGURE 55.22. Overview of the runtime architecture of applications having an ambient dynamically distributed User Interface, relying on the treatment of environment devices as remote wireless UI micro-services, over ad-hoc short-range networking.

– GUI server, such as the device offers a full-fledged remote GUI API3

The software architecture illustrated in Fig. 55.22 reflects the following key properties of ambient interactions, as they can be implemented through the Voyager development framework:

- Proximate devices are discovered dynamically, while devices getting dynamically out of the proximity range (the range can be fined tuned for Bluetooth) are lost.
- Every discovered device is communicated with a standard query protocol through which application clients (running at the PPU) may extract information regarding the hosted UI I/O resources (e.g., microservices).
- Clients may establish separate communication channels with each such distinct UI resources at ambient devices, through the control protocol, after successfully granting the UI resource for exclusive use.
- Once devices are lost (e.g., get out of the communication range) or if a previously granted resource is no longer needed, it is released for further use by other clients.

Examples of Bluetooth devices hosting multiple such UI microservices are depicted in Fig. 55.23. The MASC wristwatch de-

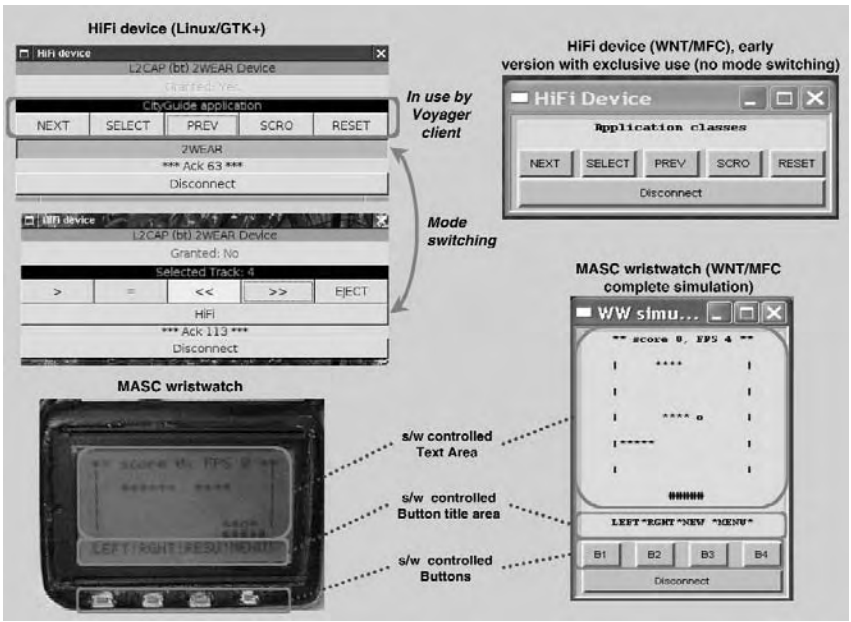


FIGURE 55.23. The Voyager-enabled wristwatch (MASC hardware version at bottom left, WNT/MFC simulation at bottom right), and the Voyager-enabled software HiFi device (Linux/GTK+ CD player at top left, WNT/MFC CD player at top right).

vices host five such basic microservices: four programmable buttons supporting text title that can be set by software (shown at bottom left) and a rectangular text area for character display. In Fig. 55.23, the wristwatch is used remotely by the Break Out game application (Savidis & Stephanidis, 2004). Environment devices, such as the Hi Fi, are entirely simulated with software running on distinct PDAs (see top left part of Fig. 55.23). The capability to make an interface running on a small PPU, using wearable Bluetooth switches for scanning interaction, while displaying remotely to wireless devices, like the wheel chair attached terminal or dynamically appearing ambient displays, relies on the following feature of Voyager: the software implementation supports input/output that is received/posted from/to remote machines over Bluetooth. Locally, within Voyager clients, the interface state and the dispatching logic are only accommodated.

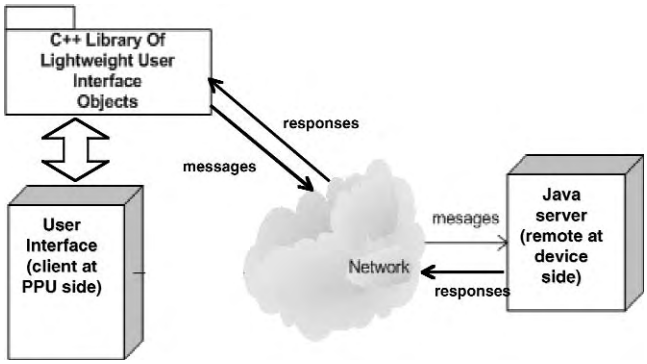


FIGURE 55.24. The use of the C++ proxy toolkit as a client-oriented wrapper for the control protocol with the remote (Bluetooth) GUI server, the latter entirely implemented in Java.

Remote Wireless GUI Service

A remote GUI service is discovered and granted in the same way as any other UI microservice. Additionally, like all other types of remote UI microservices, it offers an appropriate remote control protocol to clients. However, this particular control protocol encompasses significantly more functionality in comparison with the other more primitive UI resources, representative of the range of functions offered to applications clients by a typical GUI library. Because of the fact that the protocol is far more complicated in comparison to primitive UI resources, it has been wrapped up around an appropriate library, hiding all internal protocol and communication details (see Fig. 55.24). Following Fig. 55.24, a client application running on the PPU can utilize the remote GUI service using the proxy toolkit library, being actually a lightweight GUI toolkit that hides all wireless networking and protocol details.

This separation is similar to X Lib for the X Windowing System, where the X Lib merely implements the API, posting display requests to the X Server over TCP/IP. However, the X Protocol is more close to a graphics package, having severe communication requirements, a fact that turns its employment over Bluetooth to clearly impractical. For this reason, we decided to employ a higher level toolkit protocol for the communication between the proxy toolkit and the remote Java server (e.g., the responses and messages labels of Fig. 55.25). This protocol is a relatively old but highly effective recipe for remote

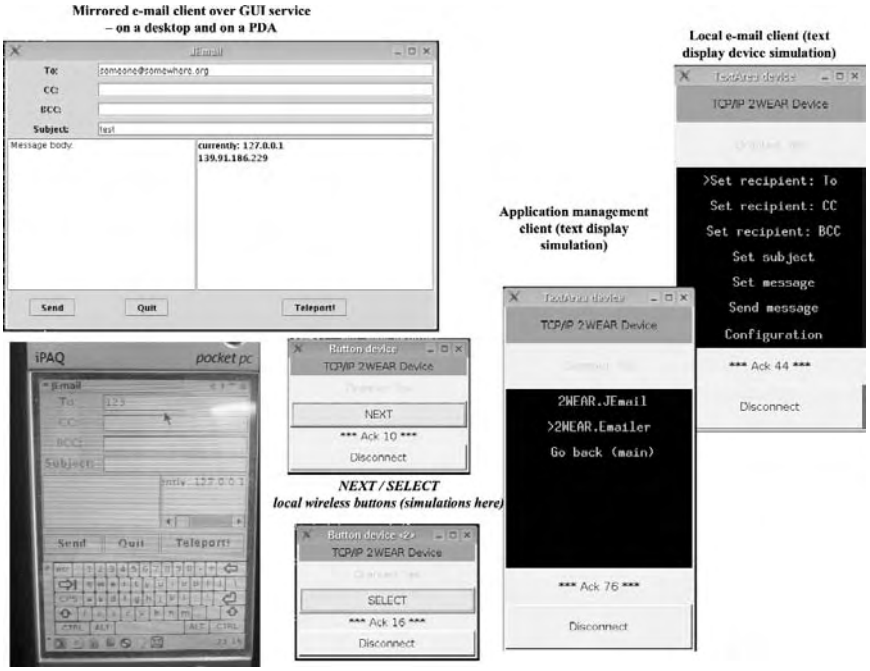


FIGURE 55.25. The different ambient devices simulated as distinct processes on different machines, communicating over Bluetooth. Note that two snapshots of the text display are shown, while there is only one in the experiment.

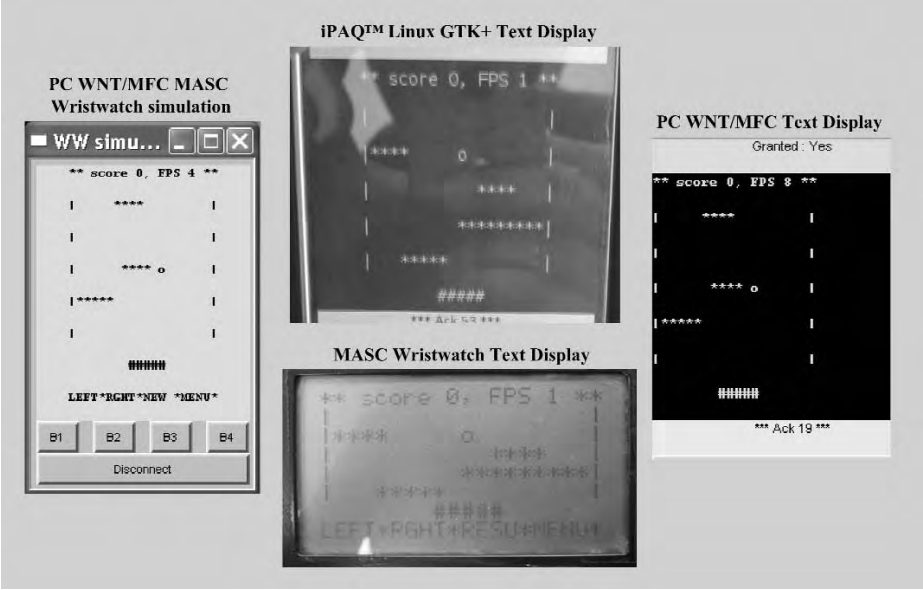


FIGURE 55.26. Alternative output configurations for the Break Out game board display. At middle bottom, the textual display of the MASC Ltd. Wristwatch is used, being a wearable Voyager enabled ambient device.

toolkit interfacing called "generic toolkit interfacing protocol" (GTIP; Savidis et al., 1997).

Test Case Applications

Various applications with an ambient user interface have been developed, including e-mail, a clock-alarm application, and a classic game remake, together with a separate special-purpose application called the "application manager." Those applications run on the PPU while displaying on the wheel chair attached text display and use the two wireless buttons. The GUI service is used dynamically to replicate the interface with a GUI instantiation (see Fig. 55.21). As it is shown, the GUI service may run on a desktop machine or even on a user PDA; this allows dynamic collaboration with a typical office able user or with a mobile user carrying a PDA device. Each distinct wireless ambient user interface microservice (e.g., the text display and the two buttons) is hosted on a different machine. The GUI service runs on a single remote machine, but supports concurrent mirroring of multiple clients.

Additionally, through the Voyager development framework (Savidis & Stephanidis, 2005), we implemented the ambient Break Out game, a variant of the famous Break Out game developed in the early 1960s, known to be the first ever action-based computer game. We have chosen this game because of its simplicity, but mainly because its graphical stages can be easily translated to textual alternatives.

Its implementation as a Voyager application has resulted in the delivery of a dynamically reconfigurable, either automatically or on user demand, user interface, both for the main game board as well as for the typical nongame dialogues. In Fig. 55.26, snapshots and photos of the alternative output configurations for the game board display are shown. The transition among such configurations is either automatic (e.g., Voyager tries to revive or to optimize the dialogue) or manual (e.g., the user updates the configuration preferences and requests explicit reconfiguration). The game configuration dialogue allows the user to choose the most preferable input and output game styles. In case the available ambient devices (proximate devices controllable remotely over Bluetooth) allow such multiple game styles to be viable, the user may switch easily on the fly among the viable alternatives, through the dialogue illustrated in Fig. 55.27. The actual PPU device where

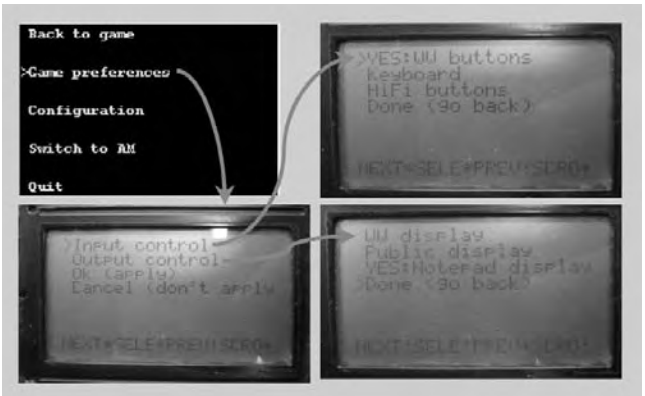


FIGURE 55.27. The game input/output style configuration dialogue. Once the "Back to game" option is chosen, Voyager will firstly try to activate the preferable pair of input and output control styles.

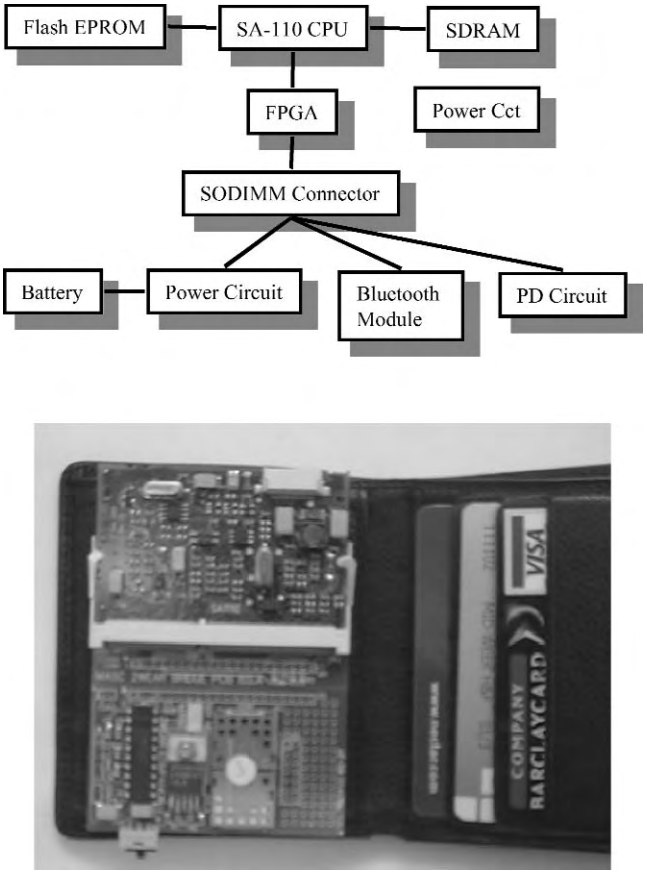


FIGURE 55.28. The pocketsize processing unit, developed by MASC Ltd for the 2WEAR project, where Voyager clients and servers run; illustration and photo are a courtesy of MASC Ltd, originally from (Majoe, 2003).

the Voyager applications and runtime components run is provided in Fig. 55.28.

Abstract Interaction Object Library and Polymorphic Alternatives

The notion of abstract dialogue objects has been formulated very early to reflect reusable logical facilities supporting basic user I/O, without, however, presupposing any particular rendering or input device policy. Early systematic work related to abstract interaction objects concerns the concept of metawidgets, representing the abstraction of interaction objects above physical platforms (Blattner, Glinert, Jorge, & Ormsby, 1992), since metawidgets are free of physical attributes and presentation issues. Similarly, the notion of virtual interaction objects has been introduced in Savidis and Stephanidis (1995), together with an interface programming language, which provided built-in support for the specification of alternative physical instantiation styles for virtual objects. In unified development, abstract interaction objects play a key role. The reengineering of existing unimorphic user interface designs, by capturing abstract interactive behaviors through a role-based model of interaction objects and retargeting the user interface physically to alternative physical forms, is a key technique of unified user interface design (Savidis & Stephanidis, 2004). This unified development technique has been also adopted in the Voyager tool, where only two abstract dialogue object classes are supported: (a) a Selector class, such as supporting user selection from an explicit list of textual options; and (b) a Text Entry class, such as enabling the user to enter textual input.

To allow programmers to program through abstract object classes, but eventually deliver full-functioning ambient dialogues the Voyager development framework employs a number of hidden implementation layers, the lower of those being responsible for direct UI microservice control. For each of the two basic object classes, the Voyager framework encapsulates appropriate runtime physical interaction control, handling the available UI services. This is accomplished through the implementation of well-defined alternative dialogue-object control policies, where each policy assumes the presence of a different group of required UI services.

Practically, each such policy delivers a way of physically implementing the interaction of its respective dialogue object with specific UI services; such policies are called "physical instantiation styles," or "styles." Input and output styles are distinguished to reflect the different ways of physical input and output binding of dialogue object classes, such as the Selector and Text Entry classes. The detailed description of the role and implementation approach for physical instantiation styles follows.

As depicted in Fig. 55.29, in the Voyager development framework, multiple implemented alternative input and output styles are provided for the Select and Text Entry dialogue objects. Each different style implements a different way of supporting input or output, requiring at runtime the presence of different UI services. Hence, each style encompasses the required UI services, as well as the algorithmic logic for using such UI services at runtime. The two dialogue object classes host the interface specific content, such as text content for selectors and input text for the Text Entry, while also implementing all the required functionality up to the point of input and output binding. The latter is formulated as follows: (a) input binding is defined as a concrete set of logical input commands, each command mapping to a corresponding function of the dialogue object class, associ-

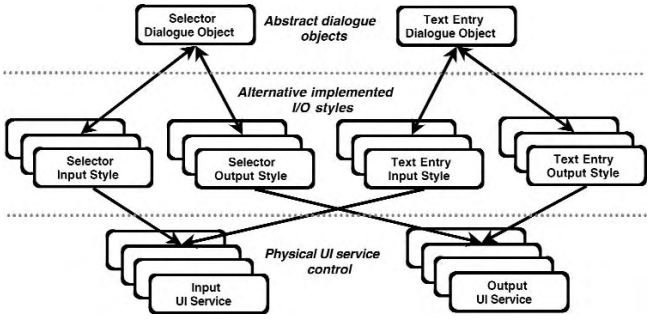


FIGURE 55.29. Implementation structure for supporting alternative I/O styles for the two abstract dialogue objects in the Voyager development framework.

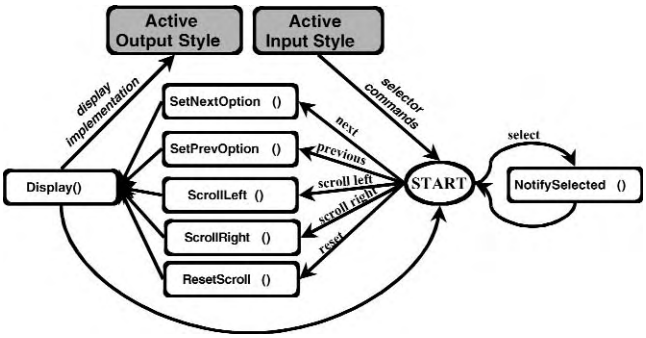


FIGURE 55.30. The main part of the dialogue automaton for the Selector dialogue object for ambient interactions.

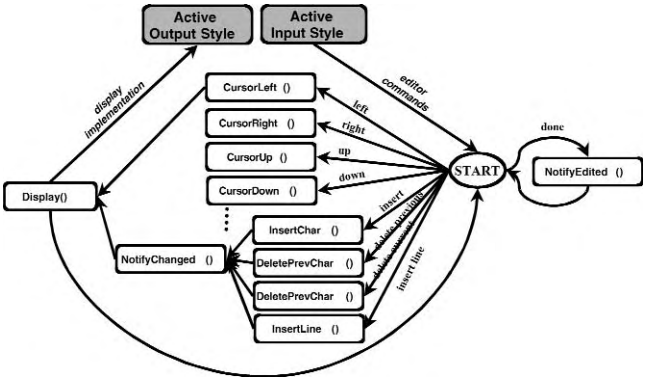


FIGURE 55.31. The main part of the dialogue automaton for the Text Entry dialogue object for ambient interactions.

ated with particular input event bindings within input style implementation; and (b) output binding is defined as an abstract function for displaying the dialogue object content, being specialized by the different output style classes.

In Fig. 55.30 and Fig. 55.31, the dialogue automata of Selector and Text Entry classes respectively are shown, indicating how the particular physical binding of the input logical dialogue commands is delegated to the implementation of the input style, and the implementation of the display function is delegated to the output style. Based on this, each distinct style may embody alternative bindings, depending on the nature of the proximate UI services dynamically discovered during user mobility and eventually granted and involved for interaction by the running applications.

CONCLUSIONS

Currently, the required technical knowledge to build user- and usage-context adapted interfaces includes user modeling, task design, cognitive psychology, rule-based systems, network communication and protocols, multiplatform interfaces, component repositories, development tools, and core user interface software engineering. Software development firms apparently prefer incremental engagement strategies, allowing a stepwise entrance to new potential markets, by delivering successive generations of products encompassing layers of novel characteristics. Similarly, the development of software applications supporting automatic user interface adaptation, for the broadest end-user population, requires a concrete strategy supporting evolutionary development, software reuse, incremental design, scalability, and modular construction. The unified user interface development discussed claims to offer a software engineering proposition that consolidates process-oriented wisdom for constructing automatically adapted user interfaces. Evolution, incremental development, and software reuse are some of the fundamental features of unified user interface development. These are reflected in the ability to progressively extend a unified user interface, by incrementally encapsulating computable content in the different parts of the architecture, to cater for additional users and usage contexts, by designing and implementing more dialogue artifacts, and by embedding new rules for the decision making logic. Such characteristics are particularly important and relevant to the claimed feasibility and viability of the proposed software engineering process and directly facilitate the practical accomplishment of universally accessible interactions.

The concept of unified user interfaces reflects a new software engineering paradigm that addresses effectively the need for interactions automatically adapted to the individual end-user requirements and the particular context of use. Following this technical approach, interactive software applications encompass the capability appropriately to deliver on the fly an adapted interface instance, performing appropriate runtime processing that engages:

- Utilization of user- and usage-context oriented information (e.g., profiles), as well as the ability to detect dynamically user- and usage-context attributes during interaction
- Management of appropriate alternative implemented dialogue components, realizing alternative ways for physical-level interaction
- Adaptation-oriented decision making that facilitates: (a) the selection, before initiation of interaction, of the most appropriate dialogue components comprising the delivered interface, given any particular dialogue context, for the particular end-user and usage-context profiles (e.g., adaptability); and (b) the implementation of appropriate changes in the initially delivered interface instance, according to dynamically detected user- and usage-context attributes (e.g., adaptivity)
- Runtime component co-ordination and control to dynamically assemble or alter the target interface; this user interface is composed on the fly from the set of dynamically selected constituent dialogue components.

The unified user interface development strategy provides a distributed software architecture with well-defined functional roles (e.g., which component does what), intercommunication semantics (e.g., which component requests what and from whom), control flow (e.g., when to do what), and internal decomposition (e.g., how the implementation of each compo-

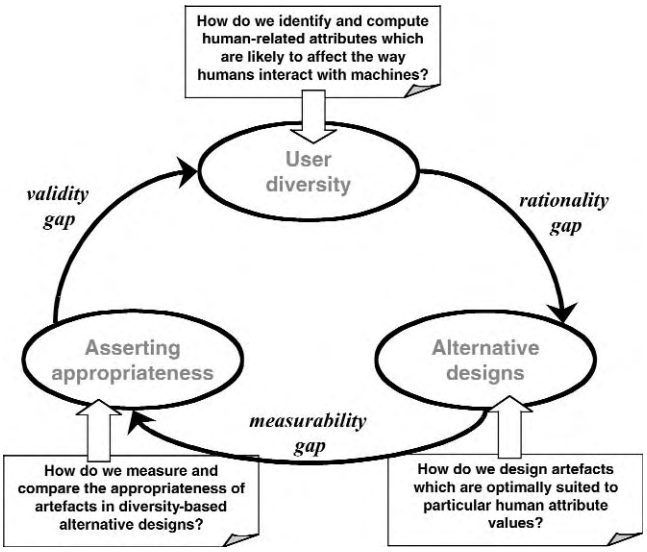


FIGURE 55.32. Open research questions in design for all.

nent is internally structured). One of the unique features of this development paradigm is the emphasis on dynamic interface assembly for adapted interface delivery, reflecting a software engineering practice with repository-oriented component organization, parametric containers with abstract containment APIs, and common interaction-monitoring control with abstract APIs. Although the method itself is not intended to be intensively prescriptive from the low-level implementation point of view, specific successful practices that have been technically validated in fieldwork regarding decision making and dynamic user-attribute detection have also been discussed, focusing on microarchitecture details and internal functional decomposition.

In this context, this development method has been systematically deployed and tested in practical situations where automatic personalization of computer-based interactive applications and services was the predominant issue. It introduces the fundamental notion of adapted interface delivery before initiation of interaction and addresses the technical challenges of coping with the inherent runtime dynamic interface assembly process.

Additionally, the basic software engineering strategy of unified development, in particular abstract objects and polymorphic containment, have been deployed in leading-edge developments, to deliver applications with ambient mobile user interfaces running on wearable pocketsize processing units.

The proposed approach establishes one possible technical route towards constructing automatically individualized best fit user interfaces: it enables incremental development and facilitates the expansion and upgrade of dialogue components as an on-going process, entailing the continuous engagement and consideration of new design parameters, and new parameter values. It is anticipated that future research work may reveal alternative approaches or methods. At the same time, further research and development work for unified user interfaces is required to address some existing challenges, mainly related to design issues (see Fig. 55.32).

Following Fig. 55.32, one top-level issue concerns the way that specific varying user attributes affecting interaction are to be identified. In other words, there is a need to identify diversity in those human characteristics that are likely to dictate alternative dialogue means. Subsequently, even if a set of those attributes is identified, it is still unclear how to conduct a design process to produce the necessary alternative dialogue artifacts for the different values of those attributes. Hence, it is necessary to design for diversity, relying upon appropriate design rationale clearly relating diverse attribute values with specific properties of the target dialogue artifacts. Currently, there is only limited knowledge about how to perform effectively this transition from alternative user-attribute values to alternative design artifacts, and it can be characterized as a rationalization gap.

Finally, the issue of how to structure appropriately alternative patterns for diverse user-attribute values, should be addressed in a way that the resulting designs are indeed efficient, effective, and satisfactory for their intended users and usage contexts. Such a process requires appropriate evaluation methods, and the capability to measure the appropriateness of designed artifacts. At present, this is still a missing link, characterized as the measurability gap. Unless we are able to assert the appropriateness of the alternative dialogue artifacts designed for diverse user attributes, we cannot validate the overall dynamically delivered interface. The inability to formulate and conduct such an evaluation process creates a validity gap.

Work currently underway, as well as future work, is expected to address these issues in an attempt to bridge the identified gaps.

ACKNOWLEDGMENTS

Most recent work regarding the Unified User Interface Development strategy has been carried out in the context of the 2WEAR—IST-2000-25286 (A Runtime for Adaptive and Extensible Wireless Wearables) project, partially funded by the Disappearing Computing Initiative (DCI) of the IST programme of the European Commission, and lasted 36 months (January 1, 2001 to December 31, 2003). The partners of the 2WEAR consortium are Foundation for Research and Technology—Hellas, Institute of Computer Science (ICS-FORTH), Greece (Prime Contractor); Nokia Research Centre (NRC), Finland; Swiss Federal Institute of Technology—Zurich Institute for Computer Systems (ETHZ), Switzerland; MA Systems and Control Limited, UK. The website of the 2WEAR Project is http://2wear.ics.forth.gr/

References

Blattner, M., Glinert, E., Jorge, J., & Ormsby, G. (1992). Metawidgets: Towards a theory of multimodal interface design. *Proceedings of COMPSAC'92 Conference, IEEE Computer Society Press* (pp. 115–120). Chicago, IL.

Hartson, H. R., Siochi, A. C., & Hix, D. (1990). The UAN: A user-oriented representation for direct manipulation interface design. ACM *Transactions on Information Systems, 8*(3), 181–203.

Hill, R. (1986). Supporting concurrency, communication and synchronisation in human-computer interaction—The Sassafras UIMS. *ACM Transactions on Graphics, 5*(3), 289–320.

Hoare, C. A. R. (1978). Communicating sequential processes. *Communications of the ACM, 21*(8), 666–677.

Johnson, P., Johnson, H., Waddington, P., & Shouls, A. (1988). Task-related knowledge structures: Analysis, modeling, and applications. In D. M. Jones, & R. Winder (Eds.), *People and computers: From research to implementation—Proceedings of HCI'88* (pp. 35–62). Cambridge, MA: Cambridge University Press.

Kobsa, A., & Pohl, W. (1995). The user modelling shell system BGP-MS. *User Modelling and User-Adapted Interaction, 4*(2), 59–106.

Majoe, D. (2003). Ubiquitous-computing enabled wireless devices. In C. Stephanidis (Ed.), *Universal access in HCI: Inclusive design in the information society*—Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Greece, 22–27 June (pp. 444–448). Mahwah, New Jersey: Lawrence Erlbaum Associates (ISBN: 0-8058-4933-5).

Marcus, A. (1996). Icon design and symbol design issues for graphical interfaces. In E. Del Galdo, & J. Nielsen (Eds.), *International user interfaces* (pp. 257–270). New York, NY: John Wiley and Sons.

Saldarini, R. (1989). Analysis and design of business information systems. In *Structured systems analysis* (pp. 22–23). New York, NY: MacMillan Publishing.

Savidis, A., & Stephanidis, C. (1995, May). Developing dual interfaces for integrating blind and sighted users: The HOMER UIMS. *Proceedings of the ACM SIGCHI 1995 Conference in Human Factors in Computing Systems* (May 7–11), (pp. 106–113). Denver, CO.

Savidis, A., Stephanidis, C., & Akoumianakis, D. (1997). Unifying toolkit programming layers: A multi-purpose toolkit integration module. In M.D. Harrison, & J.C. Torres (Eds), *Proceedings of the 4th Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '97)*, Granada, Spain, 4–6 June (pp. 177–192). Berlin: Springer-Verlag.

Savidis, A., & Stephanidis, C. (2004). Unified user interface design: Designing universally accessible interactions. *International Journal of Interacting with Computers, 16*, 243–270.

Savidis, A., & Stephanidis, C. (2005, May). Distributed interface bits: Dynamic dialogue composition from ambient computing resources. *ACM-Springer Journal on Personal and Ubiquitous Computing, 9*(3),142–168.

Stephanidis, C., Paramythis, A., Sfyrakis, M., & Savidis, A. (2001). A case study in unified user interface development: The AVANTI Web Browser. In C. Stephandis (Ed.), *User interfaces for all* (pp. 525–568). Mahwah, NJ: Lawrence Erlbaum Associates.

·56·

HCI AND SOFTWARE ENGINEERING: DESIGNING FOR USER INTERFACE PLASTICITY

Joëlle Coutaz and Gaëlle Calvary
Université Joseph Fourier, Grenoble

Introduction **1108**
The Problem Space of User Interface Plasticity . . . **1108**
Context and Context of Use 1108
Ontological Foundation for Context 1108
Observables of the Context of Use 1109
Usability 1109
System Adaptation 1111
UI Remoulding 1111
UI Distribution 1112
UI Migration 1112
State Recovery 1112
Case Studies **1112**
Home Heating Control System 1112
CamNote 1113
The Sedan-Bouillon Website 1114
Model-Driven Engineering **1114**
Models 1116
Metamodel 1116
Transformations and Mappings 1117
MDE for Plastic User Interfaces **1117**
Functional Decomposition for Plastic Interactive Systems 1118
Models for UI Plasticity 1119
M2-tasks and M2-concepts 1120
M2-Program 1121
The Models in Action 1121
Conclusion **1123**
Acknowledgments **1124**
References **1124**

INTRODUCTION

Human-computer interaction (HCI) and Software Engineering (SE) are like two old friends with different backgrounds: they share values but use them differently. Both domains address the design and development of useful and usable systems and are concerned with "requirements analysis," "incremental and iterative design," as well as "quality assurance." However, they address these problems with different development processes, different notations, and different priorities. For HCI, the human is the first-class entity in all phases of development. For SE, the final objective is a running system developed at minimal cost and delivered in time, while satisfying contractual specifications. The user is, at best, involved at the very beginning of the process, and hopefully at the very end of the project for summative evaluation. However, to avoid or correct wrong design decisions, this is too little and too late. Even in the early stages of development, functional requirements and quality goals are rarely the result of a close collaboration between HCI and SE specialists.

There are many reasons for the lack of collaboration between HCI and SE scientists and practitioners: mutual ignorance resulting from educational background, and from there, economic consideration. HCI methods such as contextual design (see chapter 49—Karen Holtzblatt), scenario-based approaches (Rosson et al., 2002), and task analysis are perceived as too demanding in terms of time and competence to inform system requirements in a formal and timely manner. On the other hand, Unified Modeling Language (UML) use cases, which express the functions that the system should support with a scenario-based flavor, are pale attempts to factor out user-centered concerns. They do not result from a human-centered requirements analysis nor do they have the expressive power of task models. Task-modeling techniques, such as Concur Task Tree (CTT) (Paternò, 2003) or User Action Notation (UAN) (Hartson, 1990), which use notations familiar to computer scientists (i.e. (LOTOS) Language of Temporal Ordering Specification operators and logic), are not used in software engineering. Conversely, domain-dependent concepts referenced in task models are ill defined, whereas UML class diagrams would improve task specifications significantly.

In summary, HCI and SE pursue the same goal, using development processes and notations that sometimes overlap and complement each other. In this chapter, we present one way to exploit both fields for the development of *plastic user interfaces* using the notion of *model* as the keystone between the two disciplines. In the following section, we present the problem space of user interface (UI) plasticity followed, by three exemplars of plastic interactive systems that illustrate aspects of the problem space. We then introduce the key objectives and principles of model-driven Engineering (http://planetmde.org) and show how they can be exploited as a solution space for UI plasticity.

THE PROBLEM SPACE OF USER INTERFACE PLASTICITY

The term *plasticity* is inspired from the capacity of solids and biological entities such as plants and brain to adapt to external constraints to preserve continuous usage. Applied to interactive systems, UI plasticity is the capacity of user interfaces to adapt to the context of use while preserving usability (Thevenin & Coutaz, 1999). We define these terms (context of use, usability, and system adaptation) in detail in the following sections.

Context and Context of Use

It is commonly agreed that *context* is about evolving, structured, and shared information spaces, and that such spaces are designed to serve a particular purpose (Coutaz et al., 2005). In other words, context is not simply a state but part of a process (the purpose). Thus, there is no such thing as *the* context, but there is *a* context qualified by the process it serves. This is why we use the phrase *context of use*, and not simply the word *context*, to refer to the information spaces that serve the adaptation process when context changes. A context change could be defined as the modification of the value of any element of the contextual information spaces. This definition would lead to an explosion of contexts. We need more structure. The following ontological foundation provides this structure.

Ontological Foundation for Context

As shown in Fig. 56.1, a contextual information space is modeled as a directed graph where a node denotes a context and an edge denotes a condition to move between two contexts. In turn, a context is a directed graph of situations where a node denotes a situation and an edge denotes a condition to move between two situations. Thus, a contextual information space is a two-level data structure, i.e. a graph of contexts where each context is in turn a graph of situations. If more structure is needed, situations may in turn be refined into "subsituations," and so on. We now need to specify the domain of definition of contexts and situations.

A context is defined over a set E of *entities*, a set Ro of *roles* (i.e. functions) that these entities may satisfy, and a set Rel of *relations* between the entities. Entities, roles, and relations are modeled as expressions of *observables* that are captured and inferred by the system. For example, in a conference room, E denotes the participants, Ro denotes the roles of speaker and listener, and Rel denotes some spatial relations such as "in front of." The situations that pertain to the same context share the sets E, Ro, and Rel.

The condition to move between two contexts is one of the following: E is replaced by a different set (e.g., the set E of participants is now replaced with the set of family members), Ro has changed (e.g., the roles of speaker and listener are replaced with that of parent), or Rel has changed (e.g., in addition to spatial relationships, temporal relationships between entities now matter).

The condition to move between two situations is one of the following:

- The cardinality of the set E has changed. For example, 10 persons enter the room and are recognized by the system as participants (their observables match the characteristics and behavior of participants). If recognized as terrorists, then the

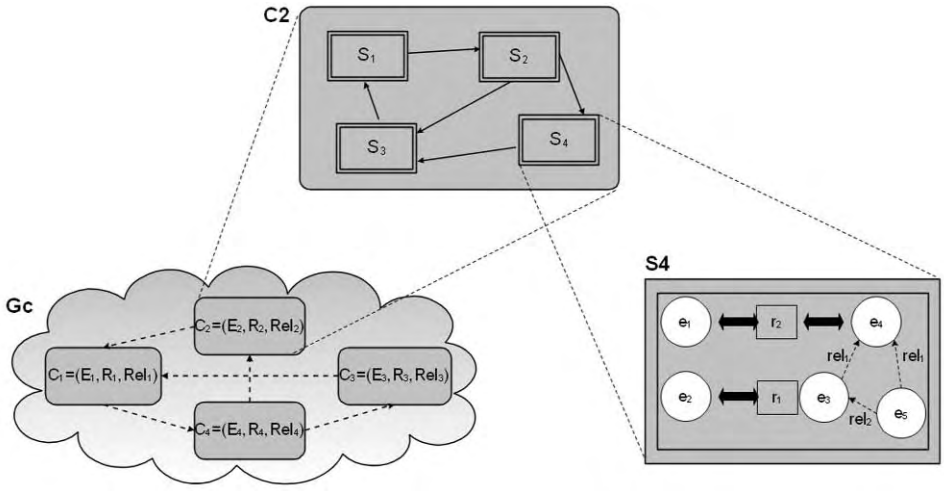


FIGURE 56.1. The graph of contexts Gc is composed of 4 contexts C_1, C_2, C_3, C_4 defined on their own sets of Entities, Roles, and Relations. In turn, Context C_2 is composed of 4 situations S_1, S_2, S_3, S_4. By definition, these situations share the same sets of Entities, Roles, and Relations. In S_4, Entities e_1 and e_4 (elements of E_2) play the role r_2 (element of R_2), whereas Role r_1 is played by Entity e_2; e_3 and e_4 satisfy Relation rel_1, e_5 and e_3 satisfy rel_2, and e_5 and e_4 are related by rel_1 (rel_1, rel_2, rel_3 are elements of Rel_2).

system would detect a context change (and not a situation change).

- A role assignment to an entity has changed (e.g., participant *e* switches from speaker to listener),
- A relation between two entities has changed (e.g., participant *e* was in front of *e'*. Now, *e'* is in front of *e*).

The ontology does not specify the nature of the entities, roles, relations, and observables. These are abstract classes from which a domain-dependent model can be specified. Using expressions of observables, designers identify the set of Entities, Roles, and Relations that are relevant for the case at hand. As discussed next, the observables of a context of use are organized into three information spaces.

Observables of the Context of Use

The observables of a context of use define three information spaces respectively called (a) the user model, (b) the environment model, and (c) the platform model.

- The user model denotes the attributes and functions that describe the archetypal person who is intended to use, or is actually using, the interactive system. This includes profile, idiosyncrasies, tasks, and activities.
- The environment model includes attributes and functions that characterize the physical places where the interaction will take place or is actually taking place. This includes numeric and/or symbolic locations (e.g., at home, in a public space, on the move in the street, a train or a car), social rules and activities, light, heat, and sound conditions.
- The platform model describes the computing, sensing, networking, and interaction resources that bind together the physical environment with the digital world.

In the conventional GUI paradigm, the platform is limited to a single computing device, typically a workstation or a PDA, connected to a network and equipped with a fixed set of interaction resources such as a screen, keyboard, and stylus. Technological advances are leading to the capacity for individuals to assemble and mould their own interactive spaces from public hot spots and private devices to access services within the global computing fabric. Interactive spaces will also take the form of autonomous computing islands, or ecosystems, whose horizon will evolve, split, and merge under human control. Figure 56.2 illustrates this view of ubiquitous computing where resources are coupled opportunistically to amplify human activities with new services, and where any real-world object (e.g., a wall, a pen and fingers) can play the role of an interaction resource. As a result, the platform is modelled as a dynamic cluster of heterogeneous resources, rather than a conventional monocomputing static device.

Usability

The term *usability* is interpreted in different ways by authors, even within the same scientific community. Usability has been identified with ease of use and learning, while excluding utility (Shackel, 1984; Nielsen, 1993). In other cases, usability is used to denote ease of use and utility, while ignoring learning ability. In software engineering, usability is considered an intrinsic property of the software product, whereas in HCI, usability is contextual: a system is not intrinsically usable or unusable. Instead, usability arises relatively to contexts of use.

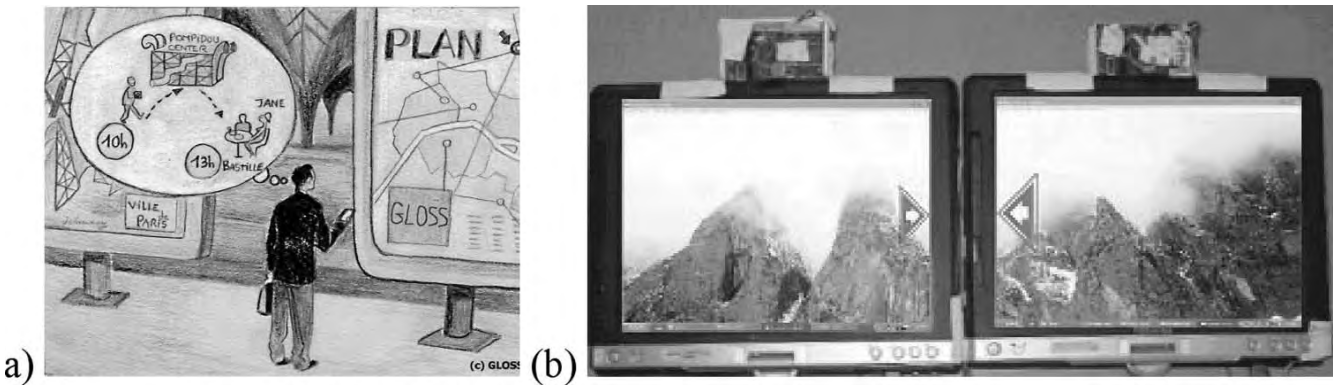


FIGURE 56.2. Assembling interaction resources opportunistically: (a) Access to services within the global computing fabric via dynamic connection of a private device to a public hot spot (e.g., an active map). (b) Ecosystem: connecting two tablets to enlarge the screen real estate, or to start an intimate collaboration between users (Hinckley, 2003).

The contextual nature of usability has been recently recognized by the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 9126 standards developed in the software community with the overarching notion of "quality in use." Quality in use is "the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety, and satisfaction in specified contexts of use." Unfortunately, as shown in Fig. 56.3, usability is viewed as one independent contribution to quality in use. Thus, the temptation is high for software people to assimilate usability to cosmetic issues limited to the user-interface component of a software product, forgetting that system latency, reliability, missing functions, and inappropriate sequencing of functions, have a strong impact on a system's "useworthiness."

Useworthiness is central to Cockton's argument for the development of systems that have value in the real world (Cockton 2004, 2005). In value-centered approaches, software design should start from the explicit expression of an intentional creation of value for a selected set of target contexts of use. Intended value for target contexts are then translated into evaluation criteria. Evaluation criteria are not necessarily elicited from generic intrinsic features such as time for task completion, but are contextualized. They are monitored and measured in real usage to assess the achieved value.

Building on Cockton's approach, we suppose that for each of the target contexts of use C_i of a system, an intended value V_i has been defined, and that V_i has been translated into the set of triples $\{(c_{i1}, d_{i1}, w_{i1}), \ldots, (c_{ij}, d_{ij}, w_{ij}), \ldots (c_{in}, d_{in}, w_{in})\}$ where c_{ij} is an evaluation criteria, and d_{ij} and w_{ij}, the expected domain of values and relative importance of c_{ij} in C_i. As just discussed, c_{ij} may be a generic measurable feature or a customized measure that depends on the intended value in C_i. Usability U_i of the system for context C_i is evaluated against a combining function F_i on the set $\{(c_{i1}, d_{i1}, w_{i1}), \ldots, (c_{ij}, d_{ij}, w_{ij}), \ldots (c_{in}, d_{in}, w_{in})\}$ whose results is intended to lie within a domain of values D_i.

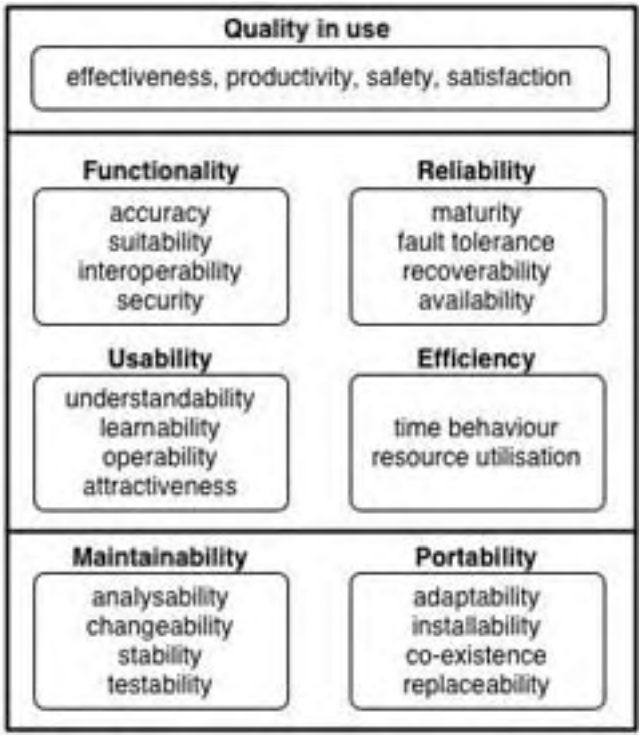


FIGURE 56.3. The usability model from ISO/IEC 9126-1.

Coming back to the notion of plasticity, an interactive system S is *plastic from a source context of use C_i to a target of use C_j* if the following two conditions are satisfied:

1. Adaptation, if needed, is supported when switching from C_i to C_j
2. Usability (value) is preserved in C_j by the adaptation process.

In turn, usability in the target context C_j is preserved if the following two conditions are satisfied:

- The usability function F_j defined for C_j lies within its intended domain D_j.
- The usability function *meta-*F_{ij} of the *meta-UI* lies within its intended domain *meta-*D_{ij} when transiting from C_i to C_j.

A meta-UI is to ambient computing what the desktop metaphor is to conventional workstations. It binds together the activities that can be performed within an interactive space. In particular, it provides users with the means to configure, control, and evaluate the adaptation process. It may, or may not, negotiate the alternatives for adaptation with the user. A *meta-UI without negotiation* makes observable the state of the adaptation process, but does not allow the user to intervene. The system is autonomous. A *meta-UI incorporates negotiation* when, for example, it cannot make sound decisions between multiple forms of adaptation, or when the user must fully control the outcome of the process. The balance between system autonomy and too many negotiation steps depends on the case at hand.

The domain of plasticity of a system is the set C of contexts of use C_i *for which usability is achieved.* We have defined usability by reasoning at the context level. If needed, a finer grain of reasoning can be applied at the situation level: intended value is defined for each situation of each context, and then translated into evaluation criteria. Preserving usability is then evaluated on situation changes.

These definitions provide a theoretical framework where value comes first and is defined on a per-context (or situation) of use basis. For each of the intended target contexts (or situations), value is operationalized into a mix of generic and customized metrics. The problem is the identification of the relevant contexts of use and situations as well as the appropriate translation of value into significant metrics. We have no answer for operationalizing value, except to use generic measures when applicable, to instrument the system appropriately using sound software development techniques, such as Aspect Oriented Programming (AOP) (Elrad, 2001), and to apply a healthy dose of common sense. On the other hand, our ontological framework on context and its associated method (Rey, 2005), can be used to define the boundaries of contexts and situations of use as well as their relationships. For our notion of context of use, the fundamental entities are the user(s), environment, and platform, each of them being characterized by observables monitored by the system. The section, "Functional Decomposition for Plastic Interactive Systems" shows how to integrate the monitoring of observables within the software architecture of an interactive system.

System Adaptation

System adaptation to context of use can take multiple forms. Here, we limit the discussion to the consequence of adaptation as perceived by users at the user interface. Adaptation can use one, or a combination of, the following techniques (a) UI remoulding, (b) UI distribution, and (c) UI migration.

UI *Remoulding*

UI remoulding denotes the reconfiguration of the user interface that results from the application of one or several transformations on all, or parts, of the user interface. These transformations can be applied at multiple levels of abstraction, and they can be intramodal, intermodal, or multimodal. These transformations include the following:

- *Suppression* of the UI components that become irrelevant in the new situation/context;
- *Insertion* of new UI components to provide access to new services relevant in the new situation/context;
- *Substitution* of UI components when UI components are replaced with new ones. Substitution can be viewed as a combination of suppression and insertion of UI components;
- *Reorganization* of UI components by revisiting their spatial layout and/or their temporal dependency. Reorganization may result from the suppression, insertion, or substitution of UI components. On the other hand, switching from a portrait view to a landscape view requires spatial reorganization only.

Remoulding is *intramodal* when the source UI components concerned by the transformations are retargeted within the same modality. Remoulding is *intermodal* if the source UI components expressed in one modality (say GUI) are transformed into UI components using a different modality (say speech). Remoulding is *multimodal* if it uses a combination of intra and intermodal transformations. For example, Teresa supports multimodal remoulding (Berti & Paternò, 2005).

The transformations can be performed at multiple levels of abstraction:

- At the *Physical Presentation* (*PP*) *level*, physical interactors (widgets) used for representing functions and concepts are kept unchanged but their rendering and behavior may change. For example, if a concept is rendered as a button class, this concept is still represented as a button in the target UI. However, the look and feel of the button or its location in the workspace may vary. This type of adaptation is used in Tk as well as in Java/AWT with the notion of peers.
- At the *Logical Presentation* (*LP*) *level*, adaptation consists of changing the representation of functions and concepts. For example, the concept of month can be rendered as a Label+Textfield, or as a Label+Combobox, or as a dedicated physical interactor. In an LP adaptation, physical interactors can replace each other provided that their representational and interactional capabilities are equivalent. The implementation of an LP-level adaptation can usefully rely on the distinction between Abstract Interactive Objects and Concrete Interactive Objects as presented in (Vanderdonckt & Bodart, 1993). Changes at the LP level imply changes at the PP level.
- At the *Dialog Component* (*DC*) *level*, the tasks that can be executed with the system are kept unchanged but their organization is modified. As a result, the structure of the dialogue structure is changed. AVANTI's polymorphic tasks (Stephanidis & Savidis, 2001) are an example of a DC-level

adaptation. Changes at the DC level imply changes at the LP and PP levels.

- At the *Functional Core Adaptor* (*FCA*) *level*, the nature of the entities as well as the functions exported by the functional core (the services) are changed. Zizi's semantic zoom is an example of an FCA-level adaptation (Zizi & Beaudouin-Lafon, 1994). Changes at the FCA level imply changes at the DC, LP, and PP levels.

UI *Distribution*

A UI is distributed when it uses interaction resources that are distributed across a cluster. For example, in graphical UI's (GUI), the rendering is distributed if it uses surfaces that are managed by different computing devices. Distribution is *static* when it is performed off-line between sessions. It is *dynamic* when it occurs on the fly.

The granularity of UI distribution may vary from application level to pixel level:

- At the *application level*, the GUI is fully replicated on the surfaces managed by each computing device. The x2vnc implementation of the VNC protocol offers an application-level distribution.
- At the *workspace level*, the unit for distribution is the workspace. A workspace is a logical space that supports the execution of a set of logically connected tasks. This concept is similar to the notion of *focus area* used in contextual design for expressing the user-environment design. Pebbles-Draw (Myers, 2001) and Rekimoto's Pick and Drop (Rekimoto, 1997) are examples of UI distribution at the workspace level.
- The *interactor level* distribution is a special case of the workspace level where the unit for distribution is an elementary interactor.
- At the *pixel level*, any user interface component can be partitioned across multiple surfaces. For example, in DynaWall (Streitz, Geibler, Holmer, Konomi, & Müller-Tomfolde. 1999), a window may simultaneously lie over two contiguous white boards as if these were managed by a single computer.

UI M*igration*

UI migration corresponds to the transfer of all or part of the UI components to different interaction resources whether these resources belong to the current platform or to another one. Migration is *static* when it is performed off-line between sessions. It is *dynamic* when it occurs on the fly. In addition, the migration of a user interface is *total* if the user interface moves entirely to a different platform (application-level migration). It is *partial* (at the workspace, interactor, or pixel levels) when a subset only of the user interface moves to different interaction resources. For example, on the arrival of a PDA, the control panels currently rendered on a whiteboard migrate to the PDA (workspace-level migration).

Migration and distribution are two distinct notions: a UI may be distributed but not migratable (static distribution). A centralized UI may migrate, but if the migration is total, it remains centralized on the target platform.

State Recovery

The granularity of state recovery characterizes the effort users must apply to carry on their activity after adaptation has occurred. State recovery can be performed at the session, task, and physical action levels:

- When the system state is saved at the *session level*, users have to restart the interactive system from scratch. They rely on the state saved by the service (functional core) level before adaptation is taking place.
- At the *task level*, the user can pursue the job from the beginning of the current interrupted task (provided that the task is attainable in the retargeted system).
- At the *physical- action level*, the user is able to carry on the current task at the exact point within the current task (provided that the task is attainable in the new version of the user interface).

The examples presented next illustrate the problem space of plastic UI's.

CASE STUDIES

The Home Heating Control System, CamNote, and Sedan-Bouillon are three examples of plastic interactive systems developed according to the MDE approach presented next. The services they provide are accessible from different types of computing devices including workstations, personal digital assistants (PDA), and mobile phones. The UI of the Home Heating Control System is kept centralized on one single device at a time, whereas for CamNote and the Sedan-Bouillon website, the UI components can be dynamically distributed and migrated across the interaction resources currently available in the interactive space. CamNote and Sedan-Bouillon differ in the technological spaces used for implementation: CamNote is Java-centric whereas Sedan-Bouillon uses PHP-MySQL Internet solutions.

Home Heating Control System

The Home Heating Control System is intended to be used (a) at home through a dedicated wall-mounted device or through a PDA connected to a wireless home-net; (b) in the office, through the web, using a standard workstation; (c) anywhere using a mobile phone or a watch.

A typical user's task consists of consulting and modifying the temperature of a particular room. Figures 56.4 and 56.5 show versions of the system for a home comprised of two rooms:

- In Figure 56.4a, the screen size is comfortable enough to display the state of all of the home thermostats.
- In Fig. 56.4b, 56.4c, and 56.4d, the screen size allows the rendering of the state of one single thermostat at a time. Thus,

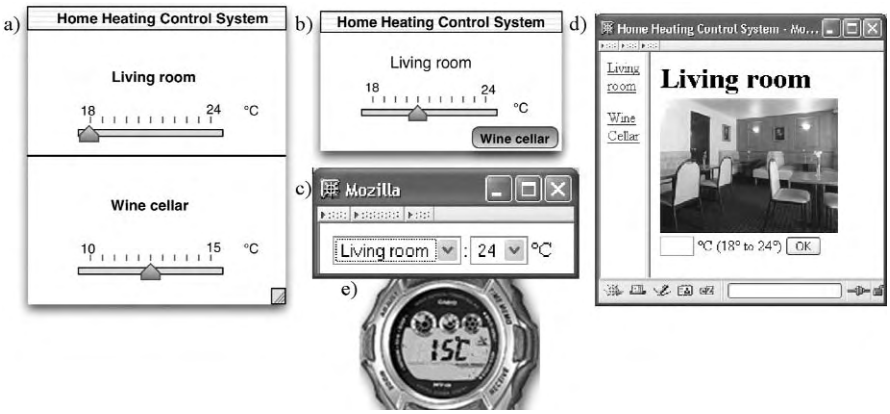


FIGURE 56.4. The Home Heating Control System. (a) Large screen. The temperature of the rooms is available at a glance. (b) (c) (d) Small screen: the temperature of a single room is displayed at a time but the system state is modifiable and browsable via a button (in b), combo-box (in c) and hyperlinks (in d). (e) Very small screen: only one single room is observable. Temperature setting and navigation tasks are not available.



FIGURE 56.5. Modifying the temperature using a mobile phone.

an additional navigation task is required to access the state of the second thermostat. In Fig. 56.4b, the task is supported by a button, whereas in Fig. 56.4c and 56.4d, a combo-box and hyperlinks are used to browse the system state. In 56.4e, the screen of the watch is so small that the user interface is limited to the living room, the most important place of the home.

Figure 56.5 shows the interaction trajectory for setting the temperature of a room with a mobile phone.

- In Fig. 56.5a, the user selects the room (e.g., "le salon"—the living room).
- In Fig. 56.5b, the system shows the current temperature of the living room.
- By selecting the editing function ("donner ordre"), one can modify the temperature of the selected room (Fig. 56.5c).

When comparing with the situation depicted in Fig. 56.4, two navigation tasks (i.e. selecting the room, and then selecting the edit function) must be performed to reach the desired state. In addition, a title has been added to every page to recall the user with the current location within the information space.

Using the taxonomy of the section "The Problem Space of User Interface Plasiticity," the Home Heating Control System can be characterized in the following way: the context of use is limited to the user's preference (i.e. the most important room in the home), and to the platform. This platform is always a monocomputing device. Context of use can be modeled as a graph of situations where the set of entities is the user and the intended computing devices. The sets of Roles and Relations are empty. Switching between two situations occurs when using a different computing device. Adaptation to context of use boils down to adapting the UI to the resources of the currently used computing device. Adaptation relies on remoulding only. It is GUI intramodal and the grain of recovery is the session: switching between devices on the fly requires starting from scratch. The adaptation level depends on the source and target platforms: when switching between the b-c-d- situations, adaptation occurs at the logical-presentation level: a widget is replaced by a functionally equivalent widget. When switching between the large screen and the small screen, the task space is not changed but organized differently: adaptation occurs at the dialogue-controller level. When switching to the watch, a number of services are no longer available: the adaptation is performed at the FCA level.

CamNote

CamNote (for CAMELEON Note) is a slides viewer that runs on a dynamic heterogeneous platform. This platform may range from a single PC to a cluster composed of a PC and a PDA. Its UI is structured into four workspaces: (a) a slides viewer, (b) a note editor for associating comments to slides, (c) a video viewer also known as "mirror pixels" that shows a live video of the speaker, and (d) a control panel to browse the slides and to setup the level of trans-

parency of the mirror. Speakers can point at items on the slide using their finger. This means of pointing is far more compelling and engaging than the conventional mouse pointer that no one can see. (Technically, the mirror is combined with the slides viewer using alpha-blending. See http://iihm.imag.fr/demos/CamNote/camnote_short.mov for a short movie demo.)

Figure 56.6a shows a configuration where the graphical UI is distributed across the screens of a PC and of a PDA. The slides viewer is displayed in a rotative canvas so that it can be oriented appropriately when projected onto a horizontal surface. If the PDA disappears, the control panel automatically migrates to the PC screen. Because different resources are now available, the control panel includes different widgets, but also a miniature representation of the speaker's video is now available. During the adaptation process, users can see the control panel emerging progressively from the slides viewer so that they can evaluate the progress of the adaptation. The UI, which was distributed on a PC and a PDA, is now centralized on the PC (Fig. 56.6b). Conversely, if the PDA reenters the interactive space, the UI automatically switches to the configuration of Fig. 56.6a, and the control panel disappears from the PC screen by weaving itself into the slides viewer before reappearing on the PDA.

In this exemplar, context of use is limited to the platform. Transitions between situations occur at the arrival or departure of a computing device. Adaptation is based on migration and distribution at the workspace level, as well as on intramodal GUI remoulding at the dialogue-controller level: when the control panel resides on the PDA, the note-editing task is no longer available. Adaptation is automatic: the user has no control over the adaptation process, but a minimum of meta-UI exists to express the transition between two situations (the weaving effect). State recovery is performed at the physical action level: the slides show is not disturbed by adaptation.

The Sedan-Bouillon Website

Sedan-Bouillon is a website that aims to promote tourism in the regions of Sedan (France) and Bouillon (Belgium) (http://www.bouillon-sedan.com/). It provides tourists with information for visiting and sojourning in these regions including a selection of hotels, camping, and restaurants. Figure 56.7a shows a simplified version of this website when a user is logged in from a PC workstation.

Preparing a trip for vacation is an exciting experience when shared by a group of people. However, one single PC screen does not necessarily favor collaborative exploration. By dynamically logging to the same website with a PDA, users are informed on the PDA that they can distribute the UI components of the site across the interaction resources currently available. In the example of Fig. 56.7b, the user asks for the following configuration: the title must appear on the PDA as well as on the PC (the title slots are ticked for the two browsers available), whereas the content should stay on the PC and the navigation bar should migrate to the PDA. Figure 56.8 shows the resulting UI. At any time, the user can ask for a reconfiguration of the UI by selecting the "meta-UI" link in the navigation bar. The UI will be reconfigured accordingly.

In terms of the plasticity problem space, the Sedan-Bouillon website is very similar to CamNote: same model of context of use, adaptation based on distribution and migration at the workspace level, and GUI intramodal remoulding at the workspace level. Contrary to CamNote, remoulding is performed at the logical-presentation level (no task is suppressed or restructured), and state recovery is supported at the task level: if adaptation occurs as the user is filling a form, the form content is lost by the adaptation process. Contrary to CamNote, the user has full control over the reconfiguration of the UI using the control panel of the meta-UI.

Having characterized three exemplars in the problem space, we now consider the method and mechanisms necessary to support interactive systems plasticity. We advocate a model-driven engineering (MDE) approach.

MODEL-DRIVEN ENGINEERING

The motivation for MDE is the integration of knowledge and techniques developed in software engineering using the notions

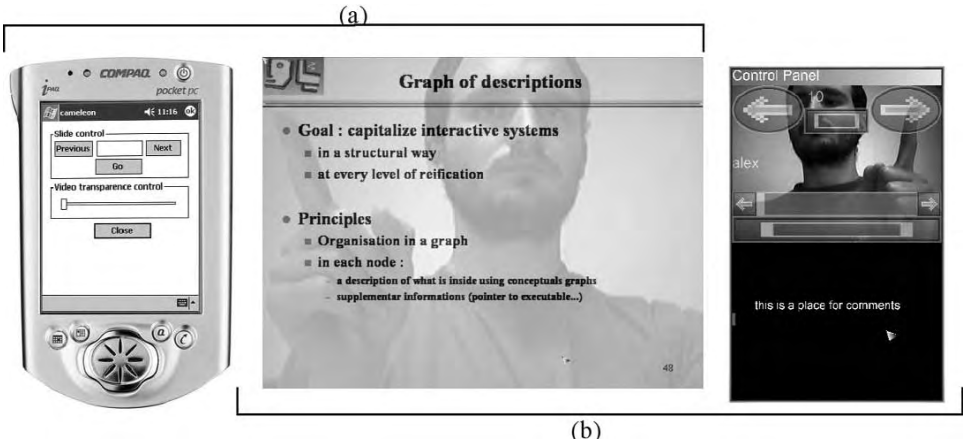


FIGURE 56.6. The user interface of CamNote. (a) The UI of CamNote when distributed on a PC and a PocketPC screens; (b) the control panel when displayed on the PC screen.

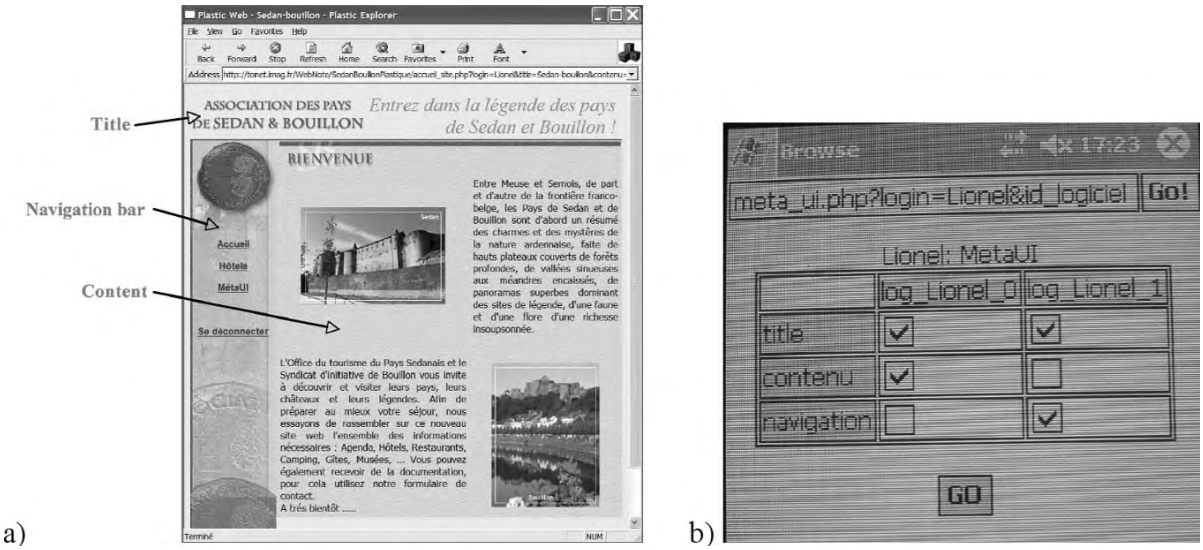


FIGURE 56.7 The Sedan-Bouillon website. (a) UI centralized on a PC screen. (b) The control panel of the meta-UI to distribute UI workspaces across the resources of the interactive space. The lines of the matrix correspond to the workspaces, and the columns denote the browsers currently used by the same user.

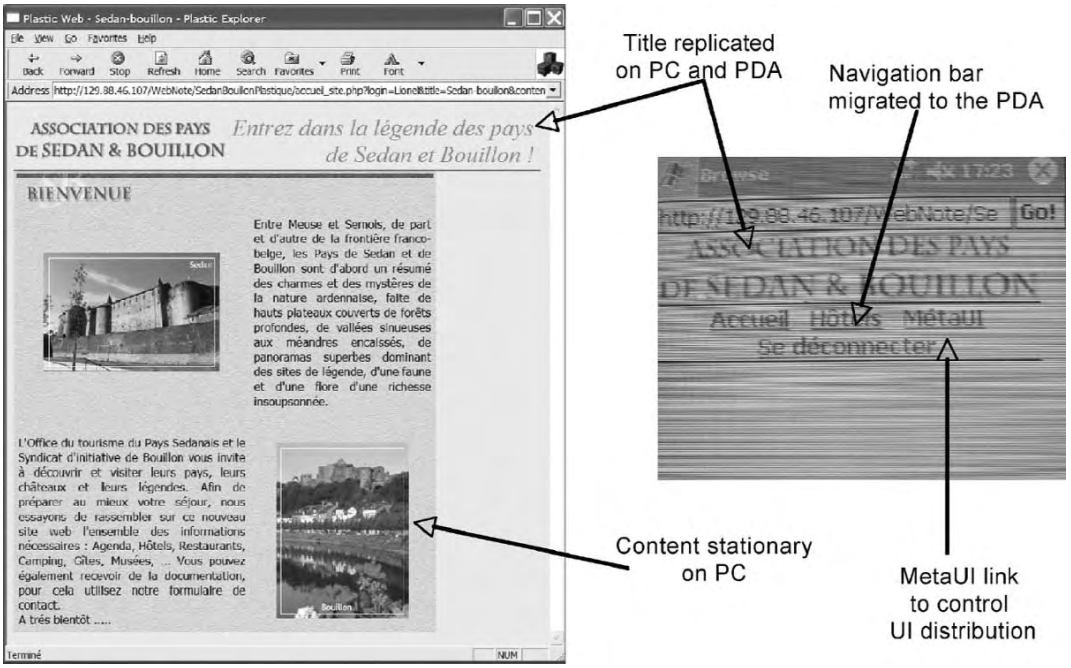


FIGURE 56.8 The Sedan-Bouillon website when distributed across the resources of the interactive space. The MetaUI link allows users to return to the configuration panel shown in Figure 7b.

of model, model transformation, and mapping as the key concepts. In the early days of computer science, software systems were simple programs written in assembly languages. In those days, a code-centric approach to software development was good enough, not to say unavoidable, to ensure a fine control over the use of computing resources. Over the years, the field has evolved into the development of distinct paradigms and application domains leading to the emergence of multiple technological spaces (TS).

"A technological space is a working context with a set of associated concepts, body of knowledge, tools, required skills, and possibilities." (Kurtev, Bézivin, & Aksit, 2002). Examples of

technological spaces include documentware concerned with digital documents using XML as the fundamental language to express specific solutions, dataware related to database systems, ontologyware, . . . HCIware!

Today, technological spaces can no longer evolve in autarky. Most of them share challenges of increasing complexity, such as adaptation, to which they can only offer partial solutions. Thus, we are in a situation where concepts, approaches, skills, and solutions need to be combined to address common problems. This is where MDE comes into play. MDE aims at achieving integration by defining gateways between technological spaces using a model-based approach. The hypothesis is that models, metamodels, model transformations, and mappings are everything.

Models

A *model* is a representation of a thing (e.g., a system), with a specific purpose. It is "able to answer specific questions in place of the actual" thing under study (Bézivin, 2004). Thus, a model, built to address one specific aspect of a problem, is by definition a simplification of the actual thing under study. For example, a task model is a simplified representation of some human activities (the actual thing under study), but it provides answers about how "representative users" proceed to reach specific goals.

A model may be *physical* (a tangible entity in the real world), *abstract* (an entity in the human mind), or *digital* (an entity within computers) (Favre, 2004a, 2004b). For example, a printed photograph of a young man named Peter is a physical representation of Peter that his mother (for example) uses for a specific purpose. Peter's mother has mental representations of him as a good son, or as a brilliant researcher (multiple abstract models about Peter). The authentification system that runs on Peter's computer knows him as a login name and password (digital model). If Peter's portrait is digitized as a JPEG picture, then the JPEG file is a digital model of a physical model. When displayed on the screen, the JPEG file is transformed into yet another digital graphics model in the system's main memory before being projected on the screen as an image (yet another physical model that Peter's mother can observe).

As this example shows, models form oriented graphs (μ graphs) whose edges denote the μ relation "is represented by". In other words, a model can represent another model, and a model can be represented by several models (see Fig. 56.9).

Models may be contemplative (not able to be processed automatically by computers) or productive (able to be processed by computers). Typically, scenarios developed in HCI (Rosson & Carroll, 2002) are contemplative models of human experience in a specified setting.

In order to be processed (by humans, and/or by computers), a model must comply with some shared syntactic and semantic conventions: it must be a well-formed expression of a language. This is true both for productive and contemplative models; most contemplative models developed in HCI use a mix of drawings and natural language. A language is the set of all well-formed expressions that comply with a grammar (along with semantics). In turn, a grammar is a model from which one can produce well-formed expressions (or models). Because a grammar is a model of a set of models, it is called a "metamodel."

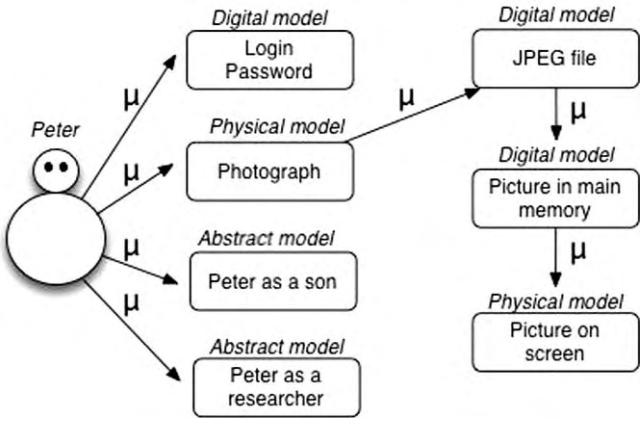


FIGURE 56.9. Models organized as oriented μ graphs (μ = "is represented by").

Metamodel

A *metamodel* is a model of a set of models that comply with it. It sets the rules for producing models. It does not represent models. Models and metamodels form a tree: a model complies to a single metamodel, whereas a metamodel may have multiple compliant models.

As an example, suppose that the authentification system mentioned above is a Java program *J*. *J* is a digital model that represents Peter and that complies with the Java grammar G_J. G_J does not represent *J*, but defines the compliance of *J* with Java. G_J is one possible metamodel, but not the only one. The authentification system could also be implemented in *C* (yet another digital model of Peter). It would then be compliant with the *C* grammar G_C. Grammars G_C and G_J could, in turn, be produced from the same grammar such as EBNF (Extended Backus-Naur Form). EBNF, defined as the ISO/IEC 14977:1996 standard, is an example of a meta-metamodel, which is model of a set of metamodels that are compliant with it. It does not represent metamodels, but sets the rules for producing distinct metamodels. As shown in Fig. 56.10, the OMG model-driven architecture (MDA) initiative has introduced a four-layer modeling stack as a way to express the integration of a large diversity of standards using MOF (meta-object facility) as the unique meta-metamodel. MDA is a specific MDE deployment effort around industrial standards including MOF, UML, CWM, QVT, etc. EBNF, G_J and G_C, the Java and C programs are models that belong to the programming technological space. Within the MDA technological space, the java source code of our authentification system becomes a UML Java model compliant with the UML metamodel. In the XML Technological Space, the Java source code could be represented as a JavaML document compliant with a JavaML DTD (document type definition). (In the XML technological space, a DTD defines the legal building blocks of an XML document.)

As shown in Fig. 56.10, the relation ("complies with") makes explicit the multiplicity of existing technological spaces as well

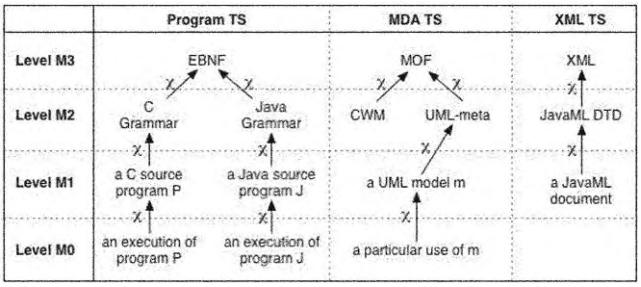


FIGURE 56.10. The OMG MDA four-layer modeling stack.

as their systematic structure into three levels of modeling spaces (the so-called M1, M2, M3 levels of MDA)+the M0 level that corresponds to a system, or parts of a system. The μ and χ relations, however, do not tell how models are produced within a technological space or how they relate to each other across distinct technological spaces. The notions of *transformation* and *mapping* is the MDE answer to this issue.

Transformations and Mappings

In the context of MDE, a *transformation* is the production of a set of target models from a set of source models, according to a transformation definition. A transformation definition is a set of transformation rules that together describe how source models are transformed into target models (Mens, Czarnecki & VanGorp, 2005). Source and target models are related by the τ relation "is transformed into." Note that a set of transformation rules is a model (a transformation model) that complies with a transformation metamodel.

Relation τ expresses an overall dependency between source and target models. However, experience shows that finer grain of correspondence needs to be expressed. Typically, the incremental modification of one source element should be propagated easily into the corresponding target element(s) and vice versa. The need for traceability between source and target models is expressed as *mappings* between source and target elements of these models. For example, each task of a task model and the concepts involved to achieve the task, are rendered as a set of interactors in the Concrete User Interface model. Rendering is a transformation where tasks and their concepts are mapped into a workspace, which, in turn, is mapped into a window populated with widgets. The correspondence between the source task (and concepts) and its target workspace, window and widgets, is maintained as a mapping function. In the conclusion, we will see how mapping functions can be exploited at run time.

Transformations can be characterized within a four-dimension space:

- The transformation may be *automated* (it can be performed by a computer autonomously), it may be *semi-automated* (requiring some human intervention), or it may be *manually performed* by a human. For example, given our current level of knowledge, the transformation of a "value-centered model" into a "usability model" can only be performed manually. On the other hand, User Interface generators such as CTTE (Mori, 2002, 2004) produce user interfaces automatically from a task model.
- A transformation is *vertical* when the source and target models reside at different levels of abstraction. UI generation is a vertical top down transformation from high-level descriptions (such as a task model) to code generation. Reverse engineering is also a vertical transformation but it proceeds bottom up, typically from executable code to some high-level representation by the way of abstraction. A transformation is *horizontal* when the source and target models reside at the same level of abstraction. For example, translating a Java source code into C code, preserves the original level of abstraction.
- Transformations are *endogenous* when the source and target models are expressed in the same language. They are *exogenous* when sources and targets are expressed in different languages while belonging to the same technological space. For example, the transformation of a Java source code program into a C program is exogenous.
- When crossing technological spaces (e.g., transforming a Java source code into a JavaML document), then additional tools (exporters or importers) are needed to bridge the gap between the spaces. *Intertechnological transformations* are key to knowledge and technical integration. This is the quest of MDE.

In the following section, we show how the MDE principles can be applied to the development of plastic interactive systems by bringing together HCIware with mainstream software technological spaces.

MDE FOR PLASTIC USER INTERFACES

There are several approaches for the development of the systems presented in the section, "Case Studies." For example, the user interface of the Home Heating Control System can be produced based on a sound iterative user-centered design process. A similar process can be conducted in parallel for the mobile phone version. In the absence of explicit links between the two streams of development, this approach is doomed to failure: inconsistencies between the two user interfaces will occur, and maintenance of the two versions will be hard to synchronize. In addition, development efforts will be duplicated.

Alternatively, the UI code for the workstation can serve as a reference for transformations into a new target UI for the mobile phone. Duplication of efforts is avoided and the approach is technically attainable. For example, using XSLT, the HTML UI produced for the workstation can easily be translated into a WML UI for the mobile phone. Crossing technological spaces at the code level is not as straightforward, but is feasible. However, this approach does not address deep UI adaptations: a code-centric approach is certainly able to address UI adaptation at the PP and LP levels (Physical and Logical Presentation). It cannot cope with higher levels of adaptation

such as DC and FCA levels (Dialogue Control and Functional Core Adaptor levels) where, for example, task sequencing may need substantial reorganization.

UI migration and UI distribution introduce new challenges. In a code-centric approach, the code of CamNote and Sedan-Bouillon website would explicitly include instructions that make reference to the context of use. For example, "If I am running on the PDA, then my rendering is *this*; if I am running on a workstation, then my rendering is *that.*". Clearly, this approach is acceptable for concept demonstration. It does not scale up in the context of ubiquitous computing where opportunism is paramount. This short analysis militates for an approach that continuously exploits models at multiple levels of abstraction not only for the development process but for the run time phase as well.

In order to define the appropriate models, we must first investigate the functions that plastic interactive spaces must support. Having identified the set of functions, we can then attach a useful set of perspectives (models) to each function.

Functional Decomposition for Plastic Interactive Systems

The picture shown in Fig. 56.11 is a reference model that makes explicit the functional decomposition of an interactive space for supporting plasticity (Balme, Demeure, Barralon, Coutaz, & Calvary, 2004). At the bottom of the picture, the hardware, a dynamic assembly of processors, sensors and actuators, as well as an heterogeneous set of *interaction resources* such as pointing devices, microphones, private-eyes, and large-size screens. Each computer executes its own *operating system* and a diversity of *virtual machines*.

The virtual machines are *model interpreters* of various sorts:

- *Java VM* is a model interpreter where the model is a bytecode program.
- *Modality interpreters* denote conventional GUI or Post-WIMP systems, speech processing as well as text-to-speech processors. They interpret models that are modality specific.
- *Other model interpreters* represent any set of interpreters used at run time including UI transformers.

The virtual machines, operating systems, and hardware (including the interaction resources) define the effective platform. The rest of the figure represents the interactive space per se. The interactive space is comprised of two functional complementary facets:

1. The fundamental functions that motivate the existence of the system.

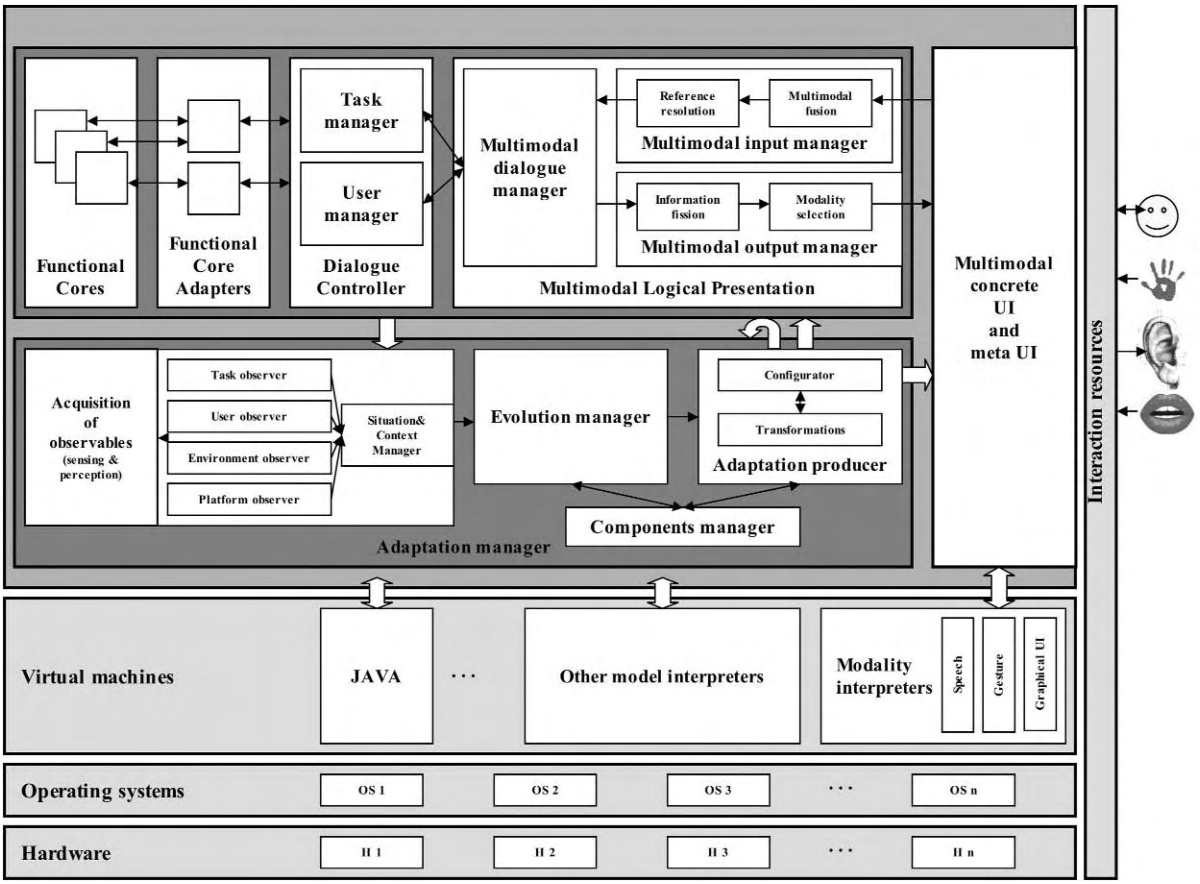


FIGURE 56.11. A reference model for the functional decomposition of plastic interactive systems.

2. The adaptation manager so that these functions can be adapted to the context of use.

The fundamental functions are structured according to the five abstraction layers of the arch reference model (Bass, Farneuf, Little, Mayer, Pellegrino, Reed, Seacord, Sheppard, & Szczur., 1992; Coutaz, 2001): (a) the Functional Core and (b) the Functional Core Adapter, (c) the Dialogue Controller, (d) the Logical and Physical Presentation layers denoted in the picture as the Multimodal Logical Presentation and Multimodal Concrete UI to stress the multimodal dimension of user interfaces (see chapter 21—Sharon Oviatt in this book for more information on multimodal interactive systems).

By acting on input interaction resources, users (represented on the figure as eyes, hands, lips, and hears), generate low-level events that are processed by the appropriate modality interpreter(s), then fed into the Multimodal Logical Presentation layer. The Multimodal Logical Presentation layer is in charge of multimodal fusion and fission both for input and output at the appropriate level of abstraction. As in the Arch model (Bass et al., 1992), the Dialogue Controller controls the domain and user-dependent task sequencing, and the Functional Core Adapters express the software interfaces to accommodate mismatches between the domain-specific Functional Cores (applications) and the user interface per se of the interactive space. This five-level Arch-based functional structure has the capacity to self-adapt to the context of use thanks to the Adaptation Manager.

The Adaptation Manager aims at satisfying three functional requirements:

1. To acquire the context of use at the appropriate level of abstraction, and to detect (and signal) context and situation changes (due, for example to the arrival of a PDA): this is performed by the Context&Situation Manager.
2. To elaborate a reaction in response to a new situation/context of use: this is performed by the Evolution Manager. For example, "If a new PDA arrives, then move the control panels of the interactive space to the PDA."
3. To execute the reaction plan: this is performed by the Adaptation Producer.

The Context&Situation Manager is structured into four levels of abstraction (Coutaz et al., 2005): at the lowest level, the system's view of the world is provided by a collection of sensors. Sensors may be physical sensors such as RFIDs, or software sensors that probe a user's identity and the state of the platform cluster. The *sensing layer* generates numeric observables. To determine meaning from numeric observables, the Context&Situation Manager must perform transformations. The *perception layer* is independent of the sensing technology and provides symbolic observables at the appropriate level of abstraction. The *situation-and-context identification layer* identifies the current situation and context from observables and detects conditions for moving between situations and contexts of use. Services in this layer specify the appropriate entities, roles, and relations for operating within the user's activities. They can be used to predict changes in situations or in contexts, and thus anticipate needs of various forms (system-centric needs as well as user-centric needs).

The reaction plan built by the Evolution Manager may require suppressing and/or replacing part or all of the Arch-based functional structure. This can involve reusing components from a *components storage*, and/or generating executable code on the fly by applying *transformations* on the models maintained by the *Models Interpreters*. Executable code is one sort of digital model to represent a particular function. In current code-centric approaches, these are the only sorts of models considered at run time. As shown in the following section, they are not the only ones in a MDE approach to UI plasticity.

Models for UI Plasticity

HCI design methods produce a large body of contemplative models such as scenarios, drawings, storyboards, and mock ups. These models are useful reference material during the design process. On the other hand, because they are contemplative, they can only be transformed manually into productive models. Manual transformation supports creative inspiration, but is prone to incorrect interpretation and loss of key information. Typically, contextual information elicited in a Contextual Design analysis, is lost during the development process. From explicit, contextual information becomes implicit within the executable program: the mapping between the real-world contextual information and its corresponding translation into code does not exist. How can we improve the situation?

Given the current state of the art in HCI and SE, we see three possible realistic bridging points:

1. Moving from contemplative scenarios to productive use cases;
2. Bringing together productive task models with UML class diagrams to cover domain-dependent services and concepts;
3. Moving from the contemplative description of context of use into the terms of the ontological space presented in the section, "Context and Context of Use."

The framework presented in Fig. 56.12 combines the second and third of our options with the MDE M1-M2 levels (Calvary, Coutaz, Thevenin, 2001; Calvary, Coutaz, Thevenin, Limbourg, Souchon, Bouillon, & Vanderdoncky, 2002). The M2 level is composed of a set of metamodels (and relations) that specify the structure of the "important" concepts of the problem space described in "The Problem of User Interface Plasticity." The M1 level is populated with multiple models that each provides an "important" perspective on a given interactive space. According to the MDE principle, the M1-level models comply with the M2 level metamodels.

Figure 56.13 makes explicit the definition of some of the M2 metamodels (and their relations) using UML as meta-metamodel. Figure 56.15 shows examples of M1-level models for the Home Heating Control System and their mapping with their respective metamodels.

The M2-models (metamodels) include, but are not limited to the following:

M2-context of use is defined as a specialization of the ontology presented in "Context and Context of Use" where users, platforms and physical environments are sorts of Entities.

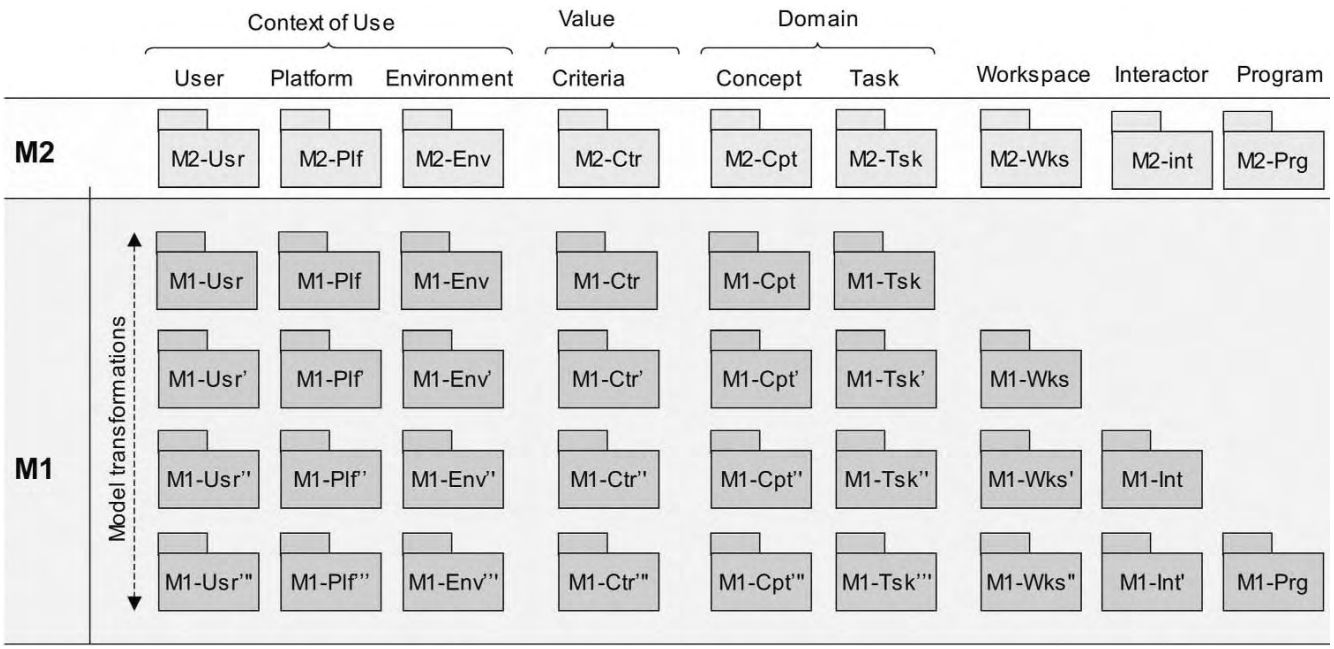


FIGURE 56.12. A Model Driven Engineering framework for plastic UI's. The set of columns represent different perspectives on a particular UI. In a column, all of the models *conform to* the same meta-model. A line corresponds to the result of a transformation step. The framework makes explicit the need for revising models during the engineering process to fit in new constraints. For instance, an initial task model (M1-Tsk) might be tuned into another version (M1-Tsk') where a task interleaving operator (Figure 4a) is replaced with a sequence operator (Figure 4b) to cope with a small screen size.

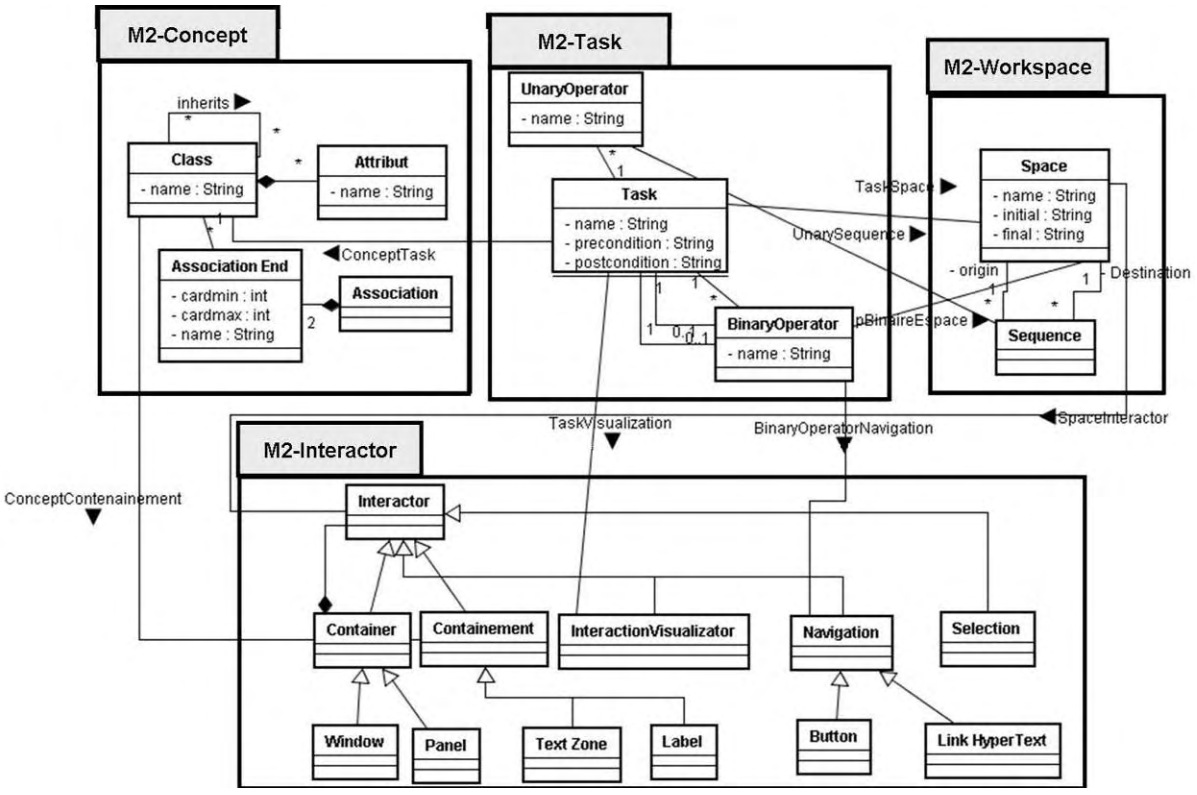


FIGURE 56.13. A subset of the meta-models of our framework for plastic UI's.

M2-evaluation criteria denotes the productive metamodel that results from a manual translation of the system value presented in "Usability."

M2-tasks and M2-concepts. As shown in Figure 56.13, a task has a name and pre- and post- conditions. It may be composed of other tasks by the way of a binary operator (such as the AND, OR, SEQ operators), or decorated with a unary operator (such as Optionality, Criticity, and Default option). BinaryOperators are related to navigation interactors to move between task spaces in the Concrete UI. A task manipulates concepts (denoted by the "ConceptTask" relation). In turn, a concept is a class composed of a set of attributes. This class may inherit from other classes, and may serve different roles. A concept is represented as interactor(s) in the Concrete UI by the way of the ConceptContainment relation. The TaskSpace relation shows how a task relates to a workspace.

M2-Workspace structures the task space at an abstract level, and *M2-interactor* describes the interactors that will populate workspaces in the Concrete UI. As shown by the definition of M2-Workspace, workspaces are chained with each other depending on the binary operator of the source tasks. A workspace is mapped into the Concrete UI as a container class interactor. This container interactor, which, in the GUI modality, may be a panel or a window, is populated with interactors that render concepts and binaryOperations (navigation operators).

M2-Program. A program is more than executable code. It covers multiple perspectives, typically: its functional decomposition, the mapping between functions and software components, the components configuration according to an architectural style, the mapping between the components and the resources of the platform (Bass, Clements, & Kazman, 1998). An architecture style is a metamodel that includes a vocabulary of design elements (e.g., pipes and filters components), that imposes configuration constraints on these elements (e.g., pipes are mono-directional connectors between two filter components), and determines a semantic interpretation that gives meaning to the system description (Garlan & Shaw, 1993). Typically, a component specifies the services (functions) it requires, the functions it provides, the quality of service it can support, the technical space to which it belongs, the "code" it encapsulates. Components as defined in distributed systems are not "rich enough" to express HCI concerns. Typically, we need to express that a component is able to support a particular set of tasks, that it can run on both a PC and a MacOS X platform, and that it supports such and such evaluation criteria in such and such situation. The Comet is one attempt in this direction (Calvary et al., 2004).

M2-Transformations correspond to the descriptions of transformations that can be automated. Figure 56.14 show examples of typical transformations between source task models and target workspaces using the following rules: (a) There must be one workspace per source task; (b) Chaining between workspaces must be based on task operators. Thus, "T= T1 T2" (which means: "to achieve T, T1 must be achieved first, followed by the achievement of T2") is transformed into two workspaces (shown in the picture as rectangles) that are accessible in sequence: the first workspace supports T1 and provides access to the workspace that supports T2. If T can be repeated, then

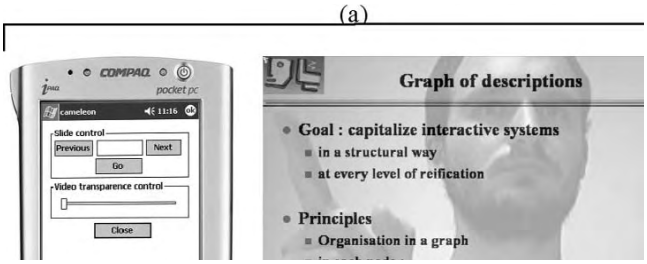


FIGURE 56.14. Typical transformation patterns between task models (expressed with UAN operators) and workspaces.

the workspace that supports T2 must provide access to the workspace that supports T1. A more formal description of this type of transformation is presented next.

In practice, how to exploit an M2-space?

The Models in Action

The M2-space offers many ways to produce plastic UIs. Let's see some examples.

The approach in conventional UI generation is to start from the production of a M1-task model and a M1-concept model, then to apply successive vertical transformations down to the generation of a M1-Program. In the example of Figure 56.15, the designer has produced the M1-Task model (top-right of the figure) and the M1-Concept model (top left) for the Home Heating Control System. These M1-level models comply with their respective M2-level models. They have been transformed by vertical top down refinement into the Concrete UI shown at the top center of Figure 56.15. This Concrete UI, as perceived and manipulated by the user, results from the interpretation of the corresponding M1-program by the model interpreters of Figure 56.11.

Because the M1-level models of our example are productive, the transformations can be performed automatically in the following way: the M1-Task model, which is comprised of one top-level task and two subtasks, is transformed into one entry/exit workspace that leads to two workspaces (bottom right of Fig. 56.15). Figure 56.16 shows the corresponding transformation description using ATL (Bézivin, Dupé, Jouault, Pitette, & Rougui, 2003) as a transformation language.

Each workspace is then mapped into a container interactor (a window for the entry/exit workspace, and two panels within the window for the other two workspaces). The concepts of M1-Concept (room and thermostats) are mapped into their respective interactors within their container interactors: on one hand, a ComboBox for specifying the room; on the other hand, a ComboBox for setting the room temperature and a label to express the unit of measure (Celsius). The interactors are then transformed into executable code. This top-down approach is used in classic UI generation. However, in classic UI generation, there is no traceability between the M1-level models. Typically, once the code is generated, the task model is lost.

In our approach, all of the M1-level models maintain an explicit mapping between them. Since they are "alive" at run time, any model can be used on the fly to inform the adaptation

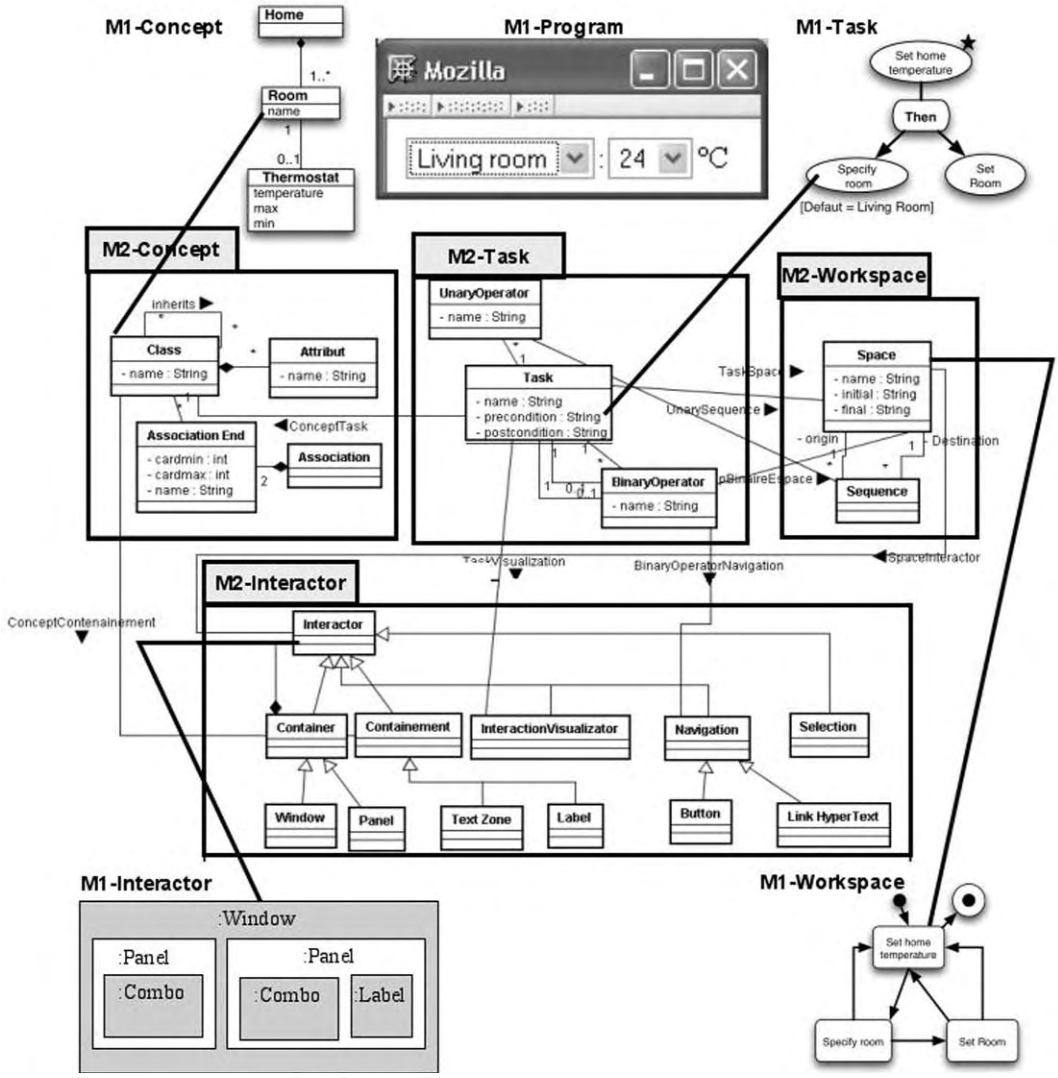


FIGURE 56.15. The Home Heating Control System from an MDE perspective. A subset of the M1-level models that correspond to the situation c) of Figure 4, and their mapping with their respective M2-level models. (For the sake of simplicity, only a subset of the mappings are represented.)

process. Thus, our approach makes it possible to combine transformations at run time, mixing bottom-up transformations to reach the appropriate level of abstraction with horizontal transformations to switch, for example, to a distinct technological space, then applying vertical refinement to produce an appropriate retargeted UI. Note that, because mapping is maintained at a fine grain, transformations can be applied only to the portions of the UI that need to be adapted.

Other forms of optimization can be performed for known recurrent situations. For example, in CamNote and the Sedan-Bouillon website, two situations have been devised by design: the dynamic arrival and departure of a PDA with the dynamic migration of workspaces between the resources of the platform. For these preplanned situations, two types of control panels have been precomputed and maintained in a repository of components. They are dynamically retrieved as needed and mapped to the appropriate computing device.

In an early version of the Home Heating Control System, switching from a UI implemented in Java/AWT to a UI implemented in WML or HTML, was not performed on the fly: the target UIs were pre-computed according to the classic top-down approach using factorization and decoration transformations at every step of the vertical refinement process. Figure 56.17 illustrates the principle of these transformations. At the top, three M1-task models have been devised for the Java/AWT, HTML and WML-enabled target platforms. The task model obtained from factorization is shown at the bottom left: 3 specific parts are linked to a shared part. The AVANTI's polymorphic tasks apply this method. On the bottom right, a unique task model is produced using decorations to express exceptions.

```
module M2TaskToM2Workspace {
  from M1Task : M2Task
  to   M1Workspace : M2Workspace
  -- One workspace per task
  rule TaskToSpace {
    from t : M2Task!Task
    to w : M2Workspace!Space (
      name <- t.name )
  }
  -- OrOperator to SequenceOperators
  rule OrOperatorToSequence{
    from o : M2Task!BinaryOperator (
      o.name = "or"
    )
    to motherToLeft : M2Workspace!Sequence (
      origin<- [ TaskToSpace.w] o.motherTask,
      destination<-[ TaskToSpace.w] o.leftTask)
```

FIGURE 56.16. An ATL transformation description based on the M2-space of Figures 56.14 and 56.16. The rule *TaskToSpace* creates one workspace *w* per source task *t* where *w* takes the name of *t*; The rule *OrOperatorToSequence* transforms all OR operators *o* between two tasks (*o.leftTask* and *o.rightTask*) into two sequence operators (from o.*motherTask* to *o.leftTask*, and *o.leftTask* to *o.rightTask*).

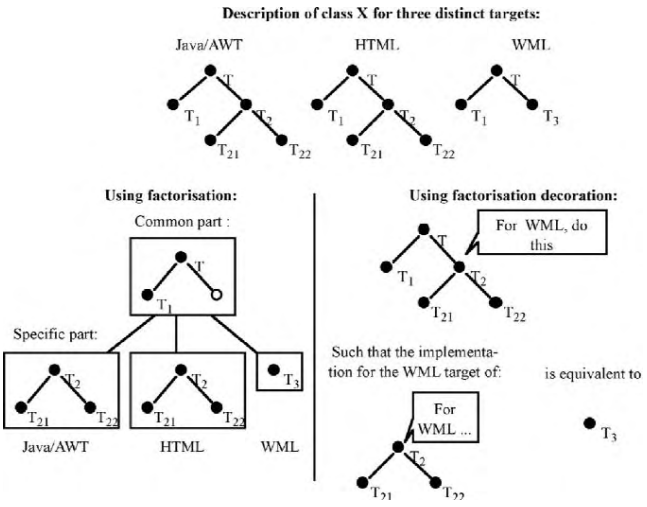


FIGURE 56.17. Factorization applied to task models.

Factorisation allows designers to focus on one target at a time, followed by the combination of the descriptions produced independently for each target into a single description where common parts are factored out. Decoration supports a different approach where designers focus on a reference target and then define exceptions.

CONCLUSION

Ad-hoc code-centric approaches to the development and deployment of plastic interactive spaces are acceptable when the intended context of use is highly constrained. In ubiquitous computing, all aspects of user-interface adaptation cannot be preprogrammed. MDE, which supports (re)computation from any level of abstraction, provides one possible avenue to address this new challenge.

In MDE, everything is a model. Models that represent a system do not float as independent things. Instead, they are organized into oriented graphs. Each graph represents an important principle of MDE: "compliance," "representation," and "transformations."

- Compliance enforces rigor. In MDE, the modeling world is organized into three levels of reasoning: the M1 model level, the M2 metalevel, and the M3 metameta level, where each level complies with the level above it.
- Representation supports the capacity to define multiple perspectives on a given thing such as an interactive space.
- Transformation and mapping are fundamental to on the fly adaptation. It also provides the formal foundation for crossing the boundaries between technological spaces. This is where SE and HCI can cross-fertilize.

In HCI, Model-based approaches to the automatic generation of user interfaces have been investigated since the mid-eighties (Cf. the seminal COUSIN (Hayes, Szekely, & Lerner, 1985) and Open-Dialogue (Schulert, Rogers, & Hamilton, 1985) User Interface Management Systems—UIMS's). Although promising, the UIMS technology has not found wide acceptance: for developers, it meant a new language to learn and designers felt severely limited by the constraints of stereotyped user interfaces (Myers, Hudson, & Pausch, 2000). The MDE-based framework we propose is intended to alleviate these problems: tools can be defined to encapsulate the low-level details of a particular language. Developers can "program" (produce models) in their favorite language since the result of this programming activity is a model that can be transformed into the appropriate target model(s). Creativity is not limited since, again, M0 level code can be vertically transformed into higher level of abstraction. The fundamental difference with the UIMS technology is that all models of a system are alive at run time: the run time system is not about executable code only.

Our MDE framework sets the global picture and principles for plastic user interfaces. It can be put in actions in many ways. Seminal work in plastic UI includes UIML (Abrams, Phanariou, Batongbacal, Williams, & Shuster, 1994) and XIML (Puerta & Eisenstein, 2001) that transform M1 level models into M0 level programs to support Logical Presentation level adaptation for centralized GUI. Tools for retargeting UI's such as Vaquita (Bouillon & Vanderdonckt, 2002) and WebRevenge (Paganelli & Paternò, 2003) correspond to a combination of bottom-up vertical, horizontal, and top-down vertical transformations. They lie within the same metameta level (the XML Technological Space), but they use distinct M2 metamodels. Vaquita and WebRevenge work off line. On the other hand, Digymes (Coninx, Luyten, Vandervelpen, Van den Bergh, & Creemers, 2003) and Icrafter (Ponnekanti, Lee, Fox, Hanrahan, & Wilograd, 2001) generate Concrete User Interfaces (CUI) at run time where a renderer dynamically computes a CUI from a workspace level model expressed in XML. Websplitter (Han, Perret, & Naghshineh, 2000)

supports the distribution of web pages content at the interactor level across the interaction resources of heterogeneous clusters, but distribution is statically specified in an XML policy file. As proof of concepts, small size exemplars have been developed for different technological spaces. The challenge is to define appropriate M2-level metamodels. Transformation is also key to the success of this approach. TransformiXML of the UsiXML (Limbourg 2004; Limbourg, Vanderdonckt, Michotte, Bouillon, & Lopez-Jacquero, 2004b) metalevel environment, which is based on graphs transformations, is certainly a promising way. Thus, there is still a long way to go. Success will require a strong active collaboration between software engineering for HCI, software architecture, machine perception, and evaluation process.

ACKNOWLEDGMENTS

This work has been supported by the Framework V FET Open GLOSS project (IST-2000-26070), CAMELEON (IST-2000-28323), and Framework VI Network of Excellence SIMILAR. The authors wish to thank Gaëtan Rey for the development of the contextor infrastructure, Lionel Balme and Alexandre Demeure for the implementation of CamNote and the Sedan-Bouillon website as well as for the development of the first version of the run-time infrastructure for plastic UI's. Thanks to Jean-Marie Favre and Jean-Sébastien Sottet for their insights into MDE principles.

References

Abrams, M., Phanariou, C., Batongbacal, A., Williams, S., & Shuster, J. (1994). UIML: an appliance-independent XML User Interface Language. *Proceedings of the 8th WWW Conference, WWW'94*.

Balme, L., Demeure, A., Barralon, N., Coutaz, J., & Calvary, G. (2004). CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces, Lecture Notes in Computer Science, Volume 3295, 2004. In P. Markopoulos, B. Eggen, E. Aarts et al. (Eds), *Ambient intelligence: Second European Symposium, EUSAI 2004*. Springer-Verlag Heidelberg (Publisher), ISBN: 3-540-23721-6, Eindhoven, The Netherlands, November 8–11, 291–302.

Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., Seacord, R., Sheppard, S., & Szczur, M. (1992). Arch, a metamodel for the runtime architecture of an interactive system. The UIMS Developers Workshop. *SIGCHI Bulletin, 24*(1), ACM Publ., 32–37.

Bass, L., Clements, P., & Kazman, R. (1998). *Software architecture in practice.* Addison Wesley Publ., ISBN 0-201-19930-0.

Berti, S., & Paternò F. (2005). Migratory multimodal interfaces in multidevice environments. In *Proceedings International Conference on Multimodal Interfaces* (*ICMI 05*)*, ACM Publ., 92–99.*

Bézivin, J., Dupé, G., Jouault, F., Pitette, G., & Rougui, J. (2003). *First experiments with the ATL transformation language: transforming XSLT into Xquery.* OOPSLA Workshop, Anaheim California USA.

Bézivin, J. (2004). In search of a basic principle for model driven engineering. *Novatica Journal, Special Issue*, March–April 2004.

Bouillon, L., & Vanderdonckt, J. (2002). Retargeting web pages to other computing platforms. *Proceedings of IEEE 9th Working Conference on Reverse Engineering WCRE'2002* (Richmond, 29 October–1 November 2002), IEEE Computer Society Press, Los Alamitos, 339–348.

Calvary, G., Coutaz, J., & Thevenin, D. (2001). A Unifying reference framework for the development of plastic user interfaces. *Proceedings of 8th IFIP International Conference on Engineering for Human-Computer Interaction EHCI'2001* (Toronto, 11–13 May 2001), R. Little and L. Nigay (eds.), Lecture Notes in Computer Science, Vol. 2254, Springer-Verlag, Berlin, 173–192.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., & Vanderdonckt, J. (2002). Plasticity of user interfaces: A revised reference framework. *First International Workshop on Task Models and Diagrams for User Interface Design TAMODIA'2002*. Bucarest, 127–134.

Calvary, G., Coutaz, J., Dâassi, O., Balme, L., & Demeure, A. (2004). Towards a new generation of widgets for supporting software plasticity: the « comet », EHCI-DSVIS'2004, *The 9th IFIP Working Conference on Engineering for Human-Computer Interaction Jointly with The 11th International Workshop on Design, Specification and Verification of Interactive Systems*. Bastide, R., Palanque, P., Roth, J. (Eds), Lecture Notes in Computer Science 3425, Springer, ISSN 0302-9743, Hamburg, Germany, 306–323.

Cockton, G. (2004). From quality in use to value in the world. In *ACM Proceedings CHI 2004,* Late Breaking Results, 1287–1290.

Cockton, G. (2005). A development framework for value-centred design. In *ACM Proceedings CHI 2005*. Late Breaking Results, 1292–1295.

Coninx, K., Luyten, K., Vandervelpen, C., Van den Bergh, J., & Creemers, B. (2003). Dygimes: Dynamically generating interfaces for mobile computing devices and embedded Systems. *In Proceedings Mobile HCI*, 256–270.

Coutaz, J. (2001). Architectural design for user interfaces. *The Encyclopedia of Software Engineering,* J. Marciniak Ed., Wiley & Sons Publ., seconde édition.

Coutaz, J., Crowley, J., Dobson, S., & Garlan, D. (2005). Context is key. *Communications of the ACM, ACM Publ., 48*(3), 49–53.

Elrad, T., Filman, R., & Bader, A. (2001). Aspect oriented programming. *Special issue, Communication of the ACM, 44*(10), 28–95.

Favre, J. M. (2004a). Foundations of model (driven) (reverse) engineering. *Dagsthul Seminar on Language Engineering for Model Driven Development, DROPS*. http://drops.dagstuhl.de/portals/04101.

Favre, J. M. (2004b). *Foundations of the meta-pyramids: Languages and metamodels*. DROPS, http://drops.dagstuhl.de/portals/04101.

Garlan, D., & Shaw, M. (1993). An Introduction to software architecture. *Advances in Software Engineering and Knowledge Engineering* 1–39. In Ambriola, V. and Tortora, G. eds., Vol. 1, World Scientific Publ., (pp. 1–39).

Han, R., Perret, V., & Naghshineh, M. (2000). WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing. *ACM Conference on Computer Supported Cooperative Work* (*CSCW 2000*), 221–230.

Hartson, R., Siochi, A., & Hix, D. (1990). The UAN: a user-oriented representation for direct manipulation interface designs. *ACM Transaction on Information Systems* (*TOIS*)*, 8*(3), 181–203.

Hayes, P J., Szekely, P., & Lerner, R. A. (1985). Design alternatives for user interface management systems based on experience with COUSIN. *In Proceedings of the ACM Conference on Human Factors in Computing Systems CHI'85.* San Francisco, CA, Apr. 14–18, 169–175.

Hinckley, K. (2003). Synchronous gestures for multiple persons and computers. *Proceedings 16th Annual ACM Symposium on User interface software and technology, UIST*, 2003, 149–158.

Kurtev, I., Bézivin, J., & Aksit, M. (2002). Technological spaces: An initial appraisal. *CoopIS, DOA'2002 Federated Conferences.* Industrial Track, Irvine.

Limbourg, Q. (2004). Multi-path development of user interfaces. PhD of University of Louvain La Neuve, Belgium.

Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., & Lopez-Jaquero, V. (2004). UsiXML: a Language Supporting Multi-Path Development of User Interfaces. *Proceedings of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems, EHCI-DSVIS'2004* (Hamburg, July 11–13, 2004).

Mens, T., Czarnecki, K. & Van Gorp, P. (2005). A taxonomy or model transformations. *Dagstuhl Seminar Proceedings 04101*. http://drops.dagstuhl.de/opus/volltexte/2005/11.

Mori, G., Paternò, F., & Santoro, C. (2002). CTTE: Support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering*, August 2002, 797–813.

Mori, G., Paternò, F., & Santoro, C. (2004). Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Transactions on Software Engineering*, August 2004.

Myers, B., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface Software Tools. *Transactions on Computer-Human Interaction* (*TOCHI*)*,* 7(1), 3–28.

Myers, B. (2001). Using handhelds and PCs together. *Communication of the ACM, 44*(11), 34–41.

Nielsen, J. (1993). *Usability engineering*. London Academic Press. ISBN 0-12-518406-9.

Paganelli, L., & Paternò, F. (2003). A tool for creating design models from website code. *International Journal of Software Engineering and Knowledge Engineering, World Scientific Publishing, 13*(2), 169–189.

Paternò, F. (2003). Concur Task Trees: An engineered notation for task models. In D. Diaper, & N. Stanton, (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 483–503 ch. 24). Lawrence Erlbaum Associates.

Ponnekanti, S., Lee, B., Fox, A., Hanrahan, P., & Winograd, T. (2001). Icrafter: A service framework for ubiquitous computing environments. In G. Abowd, B. Brumitt, S. Shafer (Eds.) *Proceedings Ubicomp 2001*, Springer Publ., LNCS 2201, 57–75.

Puerta, A. & Eisenstein, J. (2001). XIML: A common representation for interaction data. *Proceedings IUI01*, ACM publ., 214–215.

Rekimoto, J. (1997). Pick and drop: A direct manipulation technique for multiple computer environments. *In Proceedings. of UIST97, ACM Press*, 31–39.

Rey, G. (2005). *Le Contexte en interaction homme-machine: le contexteur*. PhD Thesis, Université Joseph Fourier, France.

Rosson, M. B., & Carroll, J. (2002). Usability Engineering. *Scenario-based development of human computer interaction*. Morgan Kaufmann.

Shackel, B. (1984). The concept of usability. In J. Bennett et al. (eds), *Visual display terminals: Usability issues and health concerns*, Englewood Cliffs NJ: Prentice-Hall, ISBN 0-13-942482-2.

Schulert, A. J., Rogers, G. T., & Hamilton, J. A. (1985). ADM-A dialogue manager. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (*CHI'85*), San Francisco, CA, Apr. 14–18), 177–183.

Stephanidis, C., & Savidis, A. (2001). Universal access in the information society: Methods, tools, and interaction technologies. *Journal of the Universal Access in Information Society UAIS, 1*(1), 40–55.

Streitz, N., Geibler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., & Steinmetz, R. i-LAND: An interactive Landscape for Creativity and Innovation. In *Proceedings. of the ACM conference On Human Factors in Computer Human Interaction* (*CHI99*), ACM, 120–127.

Thevenin, D., & Coutaz, J. (1999). Plasticity of User Interfaces: Framework and Research Agenda. In A. Sasse & C. Johnson (Eds.) *Proceedings Interact99*, Edinburgh, IFIP IOS Press, 110–117.

Vanderdonckt, J., & Bodard, F. (1993). Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. *Proceedings of the joint ACM Conference on Human Factors in Computing Systems CHI and IFIP Conference on Human Computer Interaction INTERACT*, April 24–29, 1993, Amsterdam, The Netherlands, ACM Press.

Zizi, M., & Beaudouin-Lafon, M. (1994). Accessing hyperdocuments through interactive dynamic maps. *Proceeding of the European Conference on Hypertext Technology ECHT*, September 19–23, 1994. Edinburgh, Scotland. ACM Press.

SECTION C

TESTING AND EVALUATION

•57•

USABILITY TESTING: CURRENT PRACTICE AND FUTURE DIRECTIONS

Joseph S. Dumas
Bentley College

Jean E. Fox
Bureau of Labor Statistics

Introduction **1130**
A Brief History of Usability Testing **1130**
Usability Testing Basics **1130**
The Focus Is on Usability 1131
The Participants Are End Users or Potential End Users 1131
There Is a Product or System to Evaluate 1131
The Participants Perform Tasks, Usually While Thinking Aloud 1131
Selecting tasks 1131
Generating task scenarios 1132
Assisting participants with tasks 1132
The Data Are Recorded and Analyzed 1133
Triangulating measures to confirm findings 1133
Identifying usability problems 1133
Assigning a severity level to problems 1133
Identifying positive issues 1133
The Results of the Test Are Communicated to Appropriate Audiences 1134
The value of highlight tapes 1134
Types of Usability Tests **1134**
Informal vs. Invalid 1134
Prototypes in Usability Tests 1135
Low-fidelity prototypes 1135
High-fidelity prototypes 1135
When to use low- and high-fidelity prototypes . . . 1135
Where Testing Occurs **1136**
Testing in a Lab 1136
What is a usability lab? 1136
Mimicking the operational environment 1136
The impact of the testing equipment 1136
One vs. multiple participants 1136
Remote Testing 1136
Onsite Testing 1137
Core Issues in Usability Testing **1138**
The Measurement of Usability 1138
Assigning severity ratings 1138
Eye tracking 1139
The Complexity and Importance of Thinking Aloud 1139
How Many Participants Are Enough? 1140
Allowing Free Exploration 1140
Reliability of Usability Testing 1141
What this means for the practitioner 1141
Validity of Usability Testing 1142
Testing With Special Populations 1142
International Participants 1142
Communication 1142
Cultural differences 1142
Disabled Participants 1143
Elderly Participants 1143
Children as Participants 1143

Testing With Special Products 1143
Mobile devices 1144
Documentation 1144
The Ethics of Usability Testing 1144
Informed consent 1144
Confidentiality of data 1144
Balancing harm with purpose 1145
Usability Testing to Show the Value of Usability 1145
Future Directions in Usability Testing 1145
Objectives Beyond Productivity 1145
Asynchronous Testing 1146
The RITE Method 1146
Conclusion 1147
References 1147

INTRODUCTION

Usability testing is one of the most popular usability methods used today, and it is the evaluation method that usability specialists say has the largest impact on product improvement (Rosenbaum, Rohn, & Humburg, 2000). Indeed, it has become so popular that some people use the term to refer to usability engineering methods in general. However, usability testing is just one of the many usability methods available for evaluating the ease of learning and use of products. This chapter will define usability testing, describe its core issues, and discuss our conception of its future directions.

A BRIEF HISTORY OF USABILITY TESTING

Usability testing gained popularity in the early 1980s, when software was beginning to reach a wider audience than just computing professionals. The explosion of end-user computing was made possible by new hardware and software in the form of both the minicomputer and microcomputer and expansion of communications technology. These advances moved computing from the isolated computer room to the desktop. The advent of the CRTs, modems, and, later in the decade, graphical user interfaces made interacting directly with the computer in real time possible.

The 1982 conference called "Human Factors in Computer Systems," held in Gaithersburg, Maryland, brought together for the first time professionals from multiple disciplines interested in studying and understanding HCI. Subsequent meetings of this group became known as the "CHI Conference." At that first meeting, a session on "Evaluating Text Editors" described what now would be called "usability tests" (Roberts & Moran, 1982). These studies were written in the style of experimental psychology reports, including sections titled "Experimental Design" and "Data Analysis," in which they described the computation of inferential statistics.

The reliance on psychological research experiments as the model for usability testing was challenged early but has persisted throughout its history. Young and Barnard (1987) proposed the concept of scenarios instead of experiments, and two years later, CHI Conference writers discussed issues such as "The role of laboratory experiments in HCI: Help, hindrance or ho-hum?" (Wolf, 1989).

The first books on HCI began appearing at this time. Perhaps the most influential book, Shneiderman's (1987) *Designing the user interface*, did not have an index item for usability testing but did have one for "Quantitative Evaluations." In that section, Shneiderman wrote:

> Scientific and engineering progress is often stimulated by improved techniques for precise measurement. Rapid progress in interactive systems design will occur as soon as researchers and practitioners evolve suitable human performance measures and techniques. . . . Academic and industrial researchers are discovering that the power of traditional scientific methods can be fruitfully employed in studying interactive systems. (p. 411)

In the second edition, there was still no entry in the index for "Usability Testing," but there was one for "Usability Laboratories." Shneiderman (1992) described usability tests but called them "pilot tests." These tests "can be run to compare design alternatives, to contrast the new system with current manual procedures, or to evaluate competitive products" (Schneiderman, 1992, p. 479). It was during this period that the first didactic books devoted to testing appeared (Dumas & Redish, 1993; Rubin, 1994).

In his third edition, there was a section in Shneiderman's (1997) book on "Usability Testing and Laboratories." Shneiderman wrote:

> Usability-laboratory advocates split from their academic roots as these practitioners developed innovative approaches that were influenced by advertising and market research. While academics were developing controlled experiments to test hypotheses and support theories, practitioners developed usability-testing methods to refine user interfaces rapidly. (p. 128)

This brief history shows that usability testing has been an established evaluation method for only about 15 years. Testing was slow to break away from its roots in psychology research—it has continued relatively unchanged until recently. However, change is in the air, as we will discuss below.

USABILITY TESTING BASICS

Valid usability tests have the following six characteristics:

- The focus is on usability.
- The participants are end users or potential end users.
- There is a product or system to evaluate.

- The participants perform tasks, usually while thinking aloud.
- The data are recorded and analyzed.
- The results of the test are communicated to appropriate audiences.

We will consider each of these basics in turn.

The Focus Is on Usability

It may seem obvious that a usability test should be about usability, but sometimes people try to use a test for other inappropriate purposes or call other methods usability tests. Perhaps the most common mismatch is between usability and marketing/promotion issues, such as adding a question to a posttest questionnaire that asks participants if they would buy the product they just used. If the purpose of the question is to provide an opportunity for participants to talk about their reactions to the test session, the question is appropriate for usability tests. However, if the question is added to predict whether customers would actually buy the product, the question is not appropriate. A six-participant usability test is not an appropriate method for estimating sales or market share. Obviously, a company would not base its sales projections on the results of the question, but people who read the test report may draw inappropriate conclusions when such a question is included.

The Participants Are End Users or Potential End Users

In a valid usability test, the participants must be part of the target market for the product. Testing with other populations may be useful in that it may find usability problems. However, the results cannot be generalized to the target population.

The key to finding people who are potential candidates for the test is a user profile (Branaghan, 1997) or a persona (Pruitt & Adlin, 2005). A user profile captures two types of characteristics:

- Those that the users share
- Those that might make a difference among users

For example, in a test of a cell phone upgrade, participants could be people who now own a cell phone or who would consider buying one. Of the people who own a phone, you may want to include people who owned the previous version of the manufacturer's phone and people who own other manufacturers' phones. You need to build a user profile of the relevant characteristics of users. It is from that profile that you create a screening questionnaire to identify potential participants for the test.

A common issue at this stage of planning is that there are more relevant groups to test than there are resources to test them. This situation forces the test team to decide which group or groups to test. This decision should be based on the product management's priorities, not on how easy it might be for the usability specialist to recruit participants.

There Is a Product or System to Evaluate

Usability testing can be performed with almost any product or technology. The range includes,

- Products with user interfaces that are all software, such as a database management system; all hardware, such as a high-quality pen; and those that have both, such as a cell phone, a clock radio, a hospital patient monitor, or a circuit board tester
- Products intended for different types of users, such as consumers, medical personnel, engineers, network managers, high-school students, computer programmers, and so forth
- Products that are used together by groups of users, such as cooperative work software (Scholtz & Bouchette, 1995)
- Products in various stages of development, such as user interface concept drawings, early low-tech prototypes, more fully functioning high-fidelity prototypes, products in beta testing, and completed products
- Components that are embedded in or accompany a product, such as print manuals, instruction sheets that are packaged with a product, tutorials, quick-start programs, online help, and so forth

The Participants Perform Tasks, Usually While Thinking Aloud

This is the execution phase of the test. It is where the test participant and the test administrator interact and it is where the data are collected. Before the test session starts, the administrator instructs the participant how the test will proceed and informs the participant that the test probes the usability of the product, not the participant's skills or experience. In many usability tests, participants are asked to think aloud, which we discuss in more detail below.

Selecting tasks. One of the essential requirements of every usability test is that the test participants attempt tasks that users of the product will perform. When a product of even modest complexity is tested, however, there are more tasks than there is time available to test them, so it is necessary to sample tasks. While not often recognized as a limitation of testing, the sample of tasks is a limitation to the scope of a test. Components of a design that are not touched by the tasks the participants perform are not evaluated. This limitation in thoroughness is often why testing is combined with usability inspection methods, which have thoroughness as one of their strengths (Cockton, Lavery, & Woolrych, 2003).

In a diagnostic test, testers select tasks for three primary reasons:

1. They include important tasks such as (a) tasks that are performed frequently or are basic to the job users will want to accomplish, and (b) tasks that are critical, if infrequent, because they affect other tasks, such as log in or configuration. With almost any product, there is a set of basic tasks. Basic means tasks that tap into the core functionality of the product. For example, nurses using a patient monitor will frequently look to see the vital sign values of the patient and

will want to silence any alarms once they determine the cause. In addition, the nurse will want to adjust the alarm limits, even though the limit adjustment may be done infrequently. Consequently, viewing vital signs, silencing alarms, and adjusting alarm limits are basic tasks.
2. They include tasks that probe areas where usability problems are likely. For example, if testers think that users will have difficulty knowing when to save their work, they may add saving work to several other tasks.
3. They include tasks that probe the components of a design, such as tasks that force the user to navigate to the lowest level of the menus or tasks that have toolbar shortcuts. The goal is to include tasks that increase thoroughness at uncovering problems.

Some additional reasons for selecting tasks are

- They may be central to the business goals for the product.
- They may be have been redesigned in response to the results of a previous test.
- They may be new to the product.

With so many reasons for selecting tasks, paring the task list to the time available is an important part of test planning. Task selection is greatly facilitated when the test team understands the tasks users want to do.

Typically, testers and developers get together in the early stages of test planning to create a list of tasks. This is a valuable opportunity to engage the product team in a discussion about usability. In addition to including a task in the list, the testers also need to make some preliminary estimate of how long each task will take. The time estimate is important for deciding how many tasks to include; it may also be useful for setting time limits for each task.

Generating task scenarios. Almost without exception, testers present the tasks that participants do in the form of a task scenario. For example:

> You have just bought a new combination telephone and answering machine. The box is on the table. Take the product out of the box and set it up so that you can make and receive calls.

From the beginning, usability testers recognized the artificiality of the testing environment. The task scenario has been an attempt to bring a flavor of the operational environment and the way the product will be used into the test. In most cases, the scenario is the only mechanism for introducing these elements into the test. Rubin (1994) described task scenarios as adding context and the participant's rationale and motivation to perform tasks. "The context of the scenarios will also help them to evaluate elements in your product's design that simply do not *jibe with reality* [italics added]," and "The closer that the scenarios represent *reality* [italics added], the more *reliable* [italics added] the test results" (Rubin, 1994, p. 125). Dumas and Redish (1999) said, "The whole point of usability testing is to predict what will happen when people use the product on their own. . . . The participants should feel as if the scenario matches what they would have to do and what they would know when they are doing that task *in their actual jobs* [italics added]" (p. 174).

The exact wording of a scenario is important. A good scenario is short, in the user's words not the product's, unambiguous, and it gives participants enough information to do the task. During test planning, testers work hard on the wording of each scenario. The goal is to make sure that the scenario probes the task the tester wants to probe.

In addition to the wording of the task scenarios, their order may also be important. It is common for task scenarios to have dependencies. For example in testing a cell phone, there may be a task to enter a phone number into memory and a later task to change it. A problem with dependencies happens when the participant cannot complete the first task. Testers should develop strategies to handle this situation, such as putting a phone number in another memory location to use when a participant could not complete the earlier task.

Testers continue to believe in the importance of scenarios and continue to use them. However, no research shows that describing tasks as scenarios rather than simple task statements makes any difference to the performance, motivation, or subjective judgments of participants. Scenarios may be essential to testing, but no research has confirmed that fact.

Assisting participants with tasks. Assisting participants to move forward in a task or a step in a task is a common practice in diagnostic tests. The primary reason for assisting is to uncover additional usability issues. Usability problems reveal themselves sequentially as a participant proceeds through a task. Consequently, if the participant can go no further because of an initial problem, subsequent problems may not be uncovered and documented. Helping the participant to move on from the initial problem is one way to uncover additional problems.

When this situation occurs, the moderator faces three issues:

1. When to intervene
2. How to provide assistance
3. What to do with the data that result

Two indicators that participants may not make any more progress are (a) participants indicate they do not know what more to do or (b) they repeat actions that do not move them toward a solution. The decision for the moderator is whether to move on to the next task or to assist. If the moderator decides that continuing with the test will provide additional diagnostic information, assistance may be warranted.

Assistance can take several forms. The moderator may initially just ask the participant to look at the task scenario again or just ask an open-ended question, such as "So what is happening here?" Sometimes these actions are enough to get the participant going again. Whether such actions constitute an assist is up to the discretion of the moderator or test team. When progress is still stalled, the moderator may then provide more specific direction, from a relatively broad statement, such as "You were getting close when you were looking in the menus," to increasingly specific directions up to telling the participant how to complete the task. The dilemma for the moderator is managing this process without providing information that will help the

participant in subsequent tasks. That kind of information may make the product look easier to use, and it may hide additional usability issues that do not reveal themselves later. The task of giving assistance is an art, and moderators must rely on their experiences and the goals of the task to determine when and how to intervene. Boren and Ramey (2000) studied methods of intervening in think aloud studies. They found that moderators at two labs intervened inconsistently. To minimize inconsistencies, the authors made several recommendations regarding when and how to intervene.

Once assists have been given, there is the additional question of how to factor them into the data analysis. For example, is a task that was completed with assistance a failed task? If so, is that always the case? In a complex product used by skilled users, some trial and error and an occasional assist might be expected and not constitute task failure. In a self-service application, the need for one assist may be enough to constitute task failure. If most participants do not require an assist to complete a task, is it fair to the design to count the task as a failure? These issues are factors in the complex decision about whether and how to change a design.

The Data Are Recorded and Analyzed

Triangulating measures to confirm findings. It is rare that a usability problem affects only one measure. For example, a poorly constructed icon toolbar will generate:

- Errors—especially picking the wrong icon on the toolbar
- Slow task times—during which participants hesitate over each icon and frequently click through them looking for the one they want
- Statements of frustration—participants express their feelings about not being able to learn how the icons are organized or be able to guess what an icon will do from the tool tip name
- The need for assistance from the moderator

Much of the data analysis involves building a case for a usability problem by combining several measures—a process that has been called "triangulation" (Dumas & Redish, 1999). The case building is driven by the problem list created during the test sessions. It is surprising how much of this analysis is dependant on the think aloud protocol. We depend on what participants say to help us understand the problems.

Identifying usability problems. Most usability problems emerge during the test itself. Testers observe usability problems during the sessions and record them on problem sheets or data logs at that time or soon after the session. Later, the problem sheet or log drives the data analysis. The sheet is typically organized by participant and by task. The test administrator records observations, such as "did not see the option," and interpretations, such as "does not understand the graphic." The test administrator notes when problems reappear.

One of the important skills in user testing is identifying the basic causes of a problem. From individual instances of problems, the experienced tester sees patterns that point to more general problems. For example, if participants look around the screen and aimlessly look through menu options, the tester might conclude that the participants were overwhelmed with the amount of information on the screen. If participants do not understand terms, the tester might conclude that the interface has too much technical and computer jargon. If participants repeat a task to make sure it was completed, the tester might conclude that there is not enough feedback about what the system is doing with the participant's actions. Seeing the underlying causes of individual problems is one of the important skills that a usability tester develops. It is not entirely clear that such skills can be taught quickly. Testers often have years of experience at studying and practicing problem identification skills.

While watching a test session, a product developer will see the same events as the test administrator. However, developers often tend to see problems as local. Instead of seeing that there needs to be a general review of the language in the interface, the developer sees problems with individual words. This conflict often does not appear until the testers and developers sit down to discuss what they saw and what to do about it. Usability professionals believe that this conflict over what really happened during the test remains a major barrier to improving a product's usability. Developers do not like hearing that they have tunnel vision and cannot see the underlying causes of individual problems, while usability professionals do not like hearing that the local fix will solve the problem. Another common barrier is a disagreement about whether participants' difficulties are caused by usability problems or problems with the methodology itself (including the selection of participants). These conflicts continue to limit the impact of testing on product improvement. However, iterative development, when it is practiced, helps resolve these conflicts, as testing subsequent iterations (perhaps with an improved methodology if needed) should reveal whether a problem still exists.

Several research studies have looked at how many usability problems are uncovered by different populations of testers. These studies consistently show that usability specialists find more problems than product developers or computer scientists (Nielsen, 1994). However, all of those studies have used inspection methods not usability testing.

Assigning a severity level to problems. One way to call attention to important problems is to put them into a measurement tool such as a problem severity scale. These scales allow testers to identify which problems are the most severe and, presumably, more likely to be fixed. In the section on measuring usability below, we discuss the validity of such scales.

Identifying positive issues. Some of the schemes for determining severity have a category for a positive issue (e.g., a good design feature). When a task is preformed smoothly and efficiently, the design must have worked. However, there has never been any attempt to categorize or rate the positive aspects of a design. Our experience is that most usability professionals pay only lip service to what works in a design, mentioning a few broad positive aspects at the beginning of a report, and then proceeding on to the long list of problems. Focusing on the negative is an aspect of human factors professionals that has been criticized from its beginnings in the 1940s (Meister, 1999). Recently, Skylar and Gilmore (2004) repeated that criticism and

called for a more positive approach. We agree that it is time for a scheme to measure the positive aspects of a design.

The Results of the Test Are Communicated to Appropriate Audiences

In the early days of user testing, there was almost always a formal report and a highlight video tape. Testers needed reports to communicate what they did and what they found. Now, it is more common for the results to be communicated more informally, such as at a meeting held soon after the last test session. Communication at these meetings is facilitated when the product team has attended at least some of the test sessions. It is also helpful if the usability specialist has been involved with the development team or at least is aware of the issues the team faced.

One of the important reasons for the change in reporting style for diagnostic usability tests is the confidence organizations have in the testing process. It is usually no longer necessary to justify the test with a written report. Organizations with active usability programs have come to accept user testing as a valid, useful evaluation tool, and an important part of their regular development processes. They do not feel that they need to know the details of the test method and the data analysis procedures. They want to know the bottom line—what problems surfaced and what should they do about them? In these organizations, a written report may still have value, but as a means of documenting and archiving the methodology rather than communicating the results.

Beginning with a project created by the National Institute of Standards and Technology (NIST), a Common Industry Format (CIF) has been created for summative tests. This format has become an ANSI standard (ANSI, 2001) and in 2005 was approved as an ISO standard. A standard format for reporting formative evaluations is in progress (Theofanos & Quesenbery, 2005). While these formats do not apply to all testing situations, they are a good start for new practitioners to learn how to create effective reports.

The value of highlight tapes. As usability testing has become an accepted evaluation tool, highlight tapes have become less common. One of the disappointing aspects of highlight tapes is that watching them does not have the same impact as seeing the sessions live. Unless the action moves quickly, even highlight tapes can be boring. This characteristic makes careful editing of the highlights a must. With the older, analog editors, it can take about one hour to create one minute of finished tape. A 15-minute tape can take two days to create, even by an experienced editor. Most of that time is taken finding appropriate segments to illustrate key findings.

The emergence of digital video has made highlight tapes less time consuming; it is still easier and more effective to have people attend the test sessions. However, it can be useful to incorporate digital video clips into presentations and reports to illustrate findings without generating a full highlight tape.

TYPES OF USABILITY TESTS

Usability testing can be helpful throughout the development lifecycle. In fact, starting early in development can help teams identify and fix usability problems as they arise, before they become too expensive to fix. Over time, investigators have refined the methodologies to better meet the objectives of the tests. Rubin (1994) identified the following types of usability tests, which all evolved from the methods used in social science research studies:

- Exploratory tests—conducted early in the development process to address high-level issues, not specific details
- Assessment tests—conducted early or midway through development; Rubin (1994) called these the "most typical" usability tests
- Validation or verification tests—conducted at the end of development to ensure that the product meets minimum usability requirements
- Comparison tests—conducted at any point in development to compare two or more products; these tests identify strengths and weaknesses of the products, which can be either competitors or competing designs for one product

Exploratory and assessment tests are diagnostic or formative tests. They help identify good designs and support improvements for bad designs. They focus on qualitative data more than quantitative data, and they rely heavily on thinking aloud procedures and prompting from the moderator. The goal of these tests is to identify usability problems (as opposed to evaluating overall usability). The methods for these tests provide investigators with flexibility to explore new issues as they arise in a test session. As such, diagnostic tests are not research studies. It is not valid to draw statistical conclusions from these tests. Investigators must use their skills to interpret descriptive statistics (e.g., four out of five participants completed the test) and qualitative data such as participant comments.

Validation/verification tests are summative tests. They are conducted near the end of development, typically with alpha or beta releases. The goal of these tests is to evaluate usability metrics (e.g., task times or error rates) to determine whether the product meets the organization's preestablished usability requirements. Thinking aloud may be discouraged, as it can interfere with the participant's performance. Comparison tests also focus on quantitative data. Validation and comparison tests both require sufficient scientific rigor to draw statistical conclusions from the data. Therefore, they still follow the standards and practices of social science experimental design, such as larger sample sizes, well-controlled tasks, and sufficient power to identify real differences. These tests are a type of research study where investigators can analyze quantitative data using inferential statistics (Dumas, 2001b).

The type of test also impacts what can be tested. Exploratory tests and assessment tests evaluate prototypes of the final product. Validation tests require a final product, or at least a near-release version, to ensure the results are valid. Comparison tests can evaluate products at any stage of development, if all versions tested are in the same stage.

Informal vs. Invalid

Diagnostic tests are sometimes called "informal" because they lack the scientific rigor of summative tests. However, it is hard

to know what informal really means. For example, Thomas (1996) described a method, called "quick and dirty" and "informal," in which the participants are not intended users of the product and in which time and other measures of efficiency are not made. Such a test may be informal, but it is certainly invalid and should not be called a usability test. It is missing one of the essentials—potential users. It is not an informal usability test because it is not a usability test at all. Despite the flexibility allowed in informal usability tests, there are still methodological requirements to ensure a test is valid. Tests that are performed quickly and with minimal resources are best called "quick and clean" rather than "informal" (Wichansky, 2000).

Prototypes in Usability Tests

Prototypes for usability can be low fidelity, high fidelity, or anywhere in between. Fidelity refers to the breadth of features implemented, the depth or degree of functionality (how complete each feature is), the look or aesthetics of the prototype, and the style of interaction (Snyder, 2004; Virzi, Sokolov, & Karis, 1996). The style of interaction refers to how users work with the prototype and how closely the medium of the prototype (e.g., paper or online) matches the final product. High-fidelity prototypes are more similar to the actual product on all these characteristics than low-fidelity prototypes are (Walker, Takayama, & Landay, 2002).

Low-fidelity prototypes. Low-fidelity prototypes are generally easy and fast to develop and can contain a surprising level of interactivity (Snyder, 2003). They can be as simple as sketches on paper. Conducting usability tests with low-fidelity prototypes requires devoting a team member to act as a facilitator, responding to the participant's input (e.g., giving the participant a new screen or menu). This person must know how the system should work to be able to respond quickly (Briggs & Johnson, 2005; Rudd, Stern, & Isensee, 1996). Snyder (2003) provides significant detail about the special concerns of using paper prototypes in usability tests.

Low-fidelity prototypes have several advantages. They can be built quickly and easily, so they allow teams to start usability testing early in development. Teams can also use low-fidelity prototypes to quickly test and resolve design conflicts throughout development. Results from usability tests on low-fidelity prototypes can be available early, allowing the team to fix design problems while it is still cost effective. Another advantage of using low-fidelity prototypes is that participants must tell the facilitator what they would do at each step. This interaction can facilitate the think aloud process, as it may seem more natural to the participants than talking to themselves (Datz-Kauffold & Henry, 2000).

Some of the challenges in working with low-fidelity prototypes include the following:

- Low-fidelity prototypes may discourage exploration of the product. Participants may not want the person acting as the computer to work harder than necessary (Snyder, 2003).
- They may be difficult for low-vision or blind participants to use if the prototype cannot accommodate a participant's usual assistive device (Briggs & Johnson, 2005).
- It is difficult to collect quantitative data if the prototype will not integrate with automated tools (Briggs & Johnson, 2005).
- Similarly, low-fidelity prototypes may not integrate with remote testing tools (Briggs & Johnson, 2005).
- Changes that the human computer makes can be more obvious than when they occur on the screen (Datz-Kauffold & Henry, 2000).

High-fidelity prototypes. High-fidelity prototypes contain more functionality and are more expensive to generate. However, there are more and more tools available to make it easier to simulate a functional product (Walker et al., 2002). In tests with high-fidelity prototypes, participants interact with the prototype just as they would with the actual product, using the same (or at least similar) hardware and software. In addition to usability tests, high-fidelity prototypes can also serve as living specifications of the product. They can communicate designs more efficiently than text documents (Rudd et al., 1996).

However, high-fidelity prototypes can give users and management the impression that the product is closer to completion that it actually is. Also, there can be features in a high-fidelity prototype the participants may expect in the final version, only to be disappointed if they are not there.

When to use low- and high-fidelity prototypes. There is some debate over whether low- or high-fidelity prototypes are more appropriate for usability testing. Generally, low-fidelity prototypes are considered most useful early in development to address high-level conceptual issues; high-fidelity prototypes are considered most useful later in development, after more design decisions have been worked out. Virzi et al. (1996) showed that low-fidelity prototypes could be useful throughout development, while others believe that low-fidelity prototypes are so different from the actual product that they are less effective (Muñoz, Miller-Jacobs, Spool, & Verplank, 1992; Rudd & Isensee, 1994).

To address this issue, several studies have compared results from usability tests of low- and high-fidelity prototypes. The research generally finds that both types find approximately the same number and types of usability problems. Walker et al. (2002) found few differences between low- and high-fidelity prototypes. They recommended reviewing the practical considerations of the test to determine the appropriate fidelity. Virzi et al. (1996) and Catani and Biers (1998) found no difference in the number and types of usability problems found with low and high fidelity prototypes. In addition, Wiklund, Thurrott, and Dumas (1992) found that the aesthetic refinement of the prototype (how finished the prototype looked) did not affect ratings of usability. Regardless of the fidelity of the prototype, it is important to remember that a prototype is not the actual product. The final product can be slower or faster. There may be features of the prototype that are not in the final product. The development team must consider which features to include in the prototype to ensure a usability test can address the appropriate issues (Virzi, 1989).

WHERE TESTING OCCURS

In this section, we consider the environment in which the testing is performed:

- Testing in a usability lab
- Remote testing
- Onsite testing

Testing in a Lab

What is a usability lab? Usability labs continue to be built, and there is a brisk business in selling lab equipment. The demand for labs is driven by the advantages of having recording equipment and the ability to allow stakeholders to view the test sessions. In essence, the method sells itself because developers and managers find the experience of watching a live test session compelling.

The basic makeup of a suite of usability test equipment has not changed much with time. It consists of video and audio recording and mixing equipment. However, some recent innovations in lab equipment have improved measurement. For example, relatively inexpensive eye tracking equipment has made it possible to know where participants are looking as they work. Also, miniaturization continues to shrink the size of almost all lab equipment. Hence, the arrival of portable lab setups that fit in airplane overhead compartments.

Labs have traditionally recorded test sessions on videotape, an analog medium. Unfortunately, the quality of analog video of the participant's computer recorded during sessions had always been poor. Second-generation copies, which are often used in highlight tapes, further degrade the quality. Scan converters selling for under $2,000 are still disappointing, making it difficult to see screen details.

However, labs are now taking advantage of digital audio and video. Screen capture software, which records the participant's computer screen digitally, has very good resolution. Labs can also capture the participant's face using Webcams, which are inexpensive and easy to use. The technology has allowed usability labs to improve recordings and to make it substantially easier to find and edit video segments. More and more tools are becoming available to capture, analyze, and store video and user actions.

Mimicking the operational environment. Testers usually try to adapt the test room to better simulate the environment in which the application or system will be used. Rubin (1994) recommended that testers "make the testing environment as realistic as possible. As much as possible, try to maintain a testing environment that mimics the actual working environment in which the product will be used" (p. 95).

The testers try to simulate a variety of characteristics in the test environment. The most obvious, and often the easiest, is making it look like the operational environment. However, other characteristics are more difficult to simulate. For example, stressors such as time pressure are difficult to mimic. Also, there are no real consequences to a participant's behavior in a test, which may impact their behavior.

Despite these concerns, the literature on product evaluation, when viewed from the perspective of 50 years, shows that researchers and practitioners have been able to use simulated complex operational environments. For example, aircraft and automobile simulators are used to study interactions with cockpits and dashboards as well as for operator training. More recently, hospital operating room simulators have been developed to study issues in anesthesiology (Gaba, 1994).

Meister (1999) emphasized the importance of understanding how the actual operational environment will impact behavior. He advocated improving our understanding of the differences between the test and operational environments through continued research. Usability testers should carefully consider the environment of each test, as both laboratory and on-site testing have their own advantages (see the discussion later in this chapter). Putting a couch in the test room to make it look more like a room in a home may not simulate the use environment, but going to the participant's home for testing is a complex process (Mitropoulos-Rundus & Muszak, 1997).

The impact of the testing equipment. An issue that has been debated throughout the history of usability testing is the impact of one-way mirrors and recording equipment on the test participants. This debate comes to a head in discussions about whether the test administrator should sit with participants as they work or stay behind the one-way mirror and talk over an intercom. Some testing groups usually sit with the participant, believing that it reduces the participants' anxiety about being in the test and makes it easier to manage the test session (Rubin, 1994). Other testing groups normally do not sit with the participants, believing that it makes it easier to remain objective and frees the administrator to record the actions of the participants (Dumas & Redish, 1999). Barker and Biers (1994) conducted an experiment in which they varied whether there was a one-way mirror or cameras in the test room. They found that the presence of the equipment did not affect the participants' performance or ratings of usability of the product. No other studies have been reported, however, on this hotly debated and important issue.

One vs. multiple participants. Most usability tests involve one test participant at a time. Studies show that when two participants work together—sometimes called the "codiscovery method" (Kennedy, 1989)—they make more utterances. The nature of the utterances also is different, with codiscovery participants making more evaluative, as opposed to descriptive statements, and making more statements that developers view as useful (Hackman & Biers, 1992). A related method is to have one participant teach another how to do a task (Vora, 1994). However, using either of these methods requires recruiting twice as many participants.

Remote Testing

A relatively recent innovation in usability testing is the ability to test when the administrator and the participant are in differ-

ent locations. There are several advantages to remote testing (Perkins, 2001):

- You can reach a worldwide population of participants because you are not limited to the local testing area.
- It is easier to get participants to volunteer because they do not have to travel.
- Participants work at their desks in their work environments, which may make them more comfortable and the testing more realistic.
- You do not need a usability lab.
- Often you do not have to compensate participants.

In the past, the technology to conduct such sessions was not good enough to allow usability specialists to get the information they need (Dumas, 2003). That is no longer true because of several factors:

- The Internet has made it possible for usability specialists and participants to work together without installing special hardware or complex software on both the tester's and the participant's computers.
- There are tools available for instrumenting websites automatically to collect usability measures and to insert questions and ratings as the participants work.
- Collaboration software that works over the Internet makes it possible to share desktops and control of the cursor.
- Recording software makes it possible to store good quality video and sound in files that are not large by today's standards, often less than 50M for a two-hour session.
- PC processors and RAM are fast enough to run both recording software and the application you are testing simultaneously. In addition, when participants are working at home, they often have broadband or high-speed transmissions allowing them to work without slow modem connections.

Remote testing takes two forms: (a) synchronous, in which the administrator and the participant work together, usually communicating over the phone, and (b) asynchronous, in which the participants work without the direct guidance of an administrator. Each of these variations has its strengths and weaknesses. In both synchronous and asynchronous testing, informed consent and nondisclosure forms need to be explained and signed by the participant.

Some remote testing configurations may present security problems. It is possible that screen shots and more can be obtained by participants without the knowledge of the test administrator. Allowing participants to share applications on computers inside your organization's firewall may be prohibited. Some organizations may be able to address this with a nondisclosure agreement, while others may require a special computer outside their firewalls.

In synchronous remote testing, the participant and tester are both involved in the test session but from different locations. They usually communicate by telephone. Therefore, synchronous testing has the same time requirements as laboratory testing, so it typically has the same small number of participants. Unless a special web camera is used, the administrator cannot see the participant. We do not yet know what the impact of not seeing the participant is, but one laboratory study indicates that usability specialists judge usability problems as less severe when they cannot see the participant's face (Lesaigle & Biers, 2000). One advantage of synchronous testing is that visitors can watch the screen activity live. If you use a phone conferencing system and multiuser screen sharing applications, they can even watch and listen from their own offices. To date, no research studies have compared synchronous remote testing and laboratory testing.

With asynchronous remote testing, participants complete the tasks on their own, and the tester reviews the session results later. Asynchronous remote testing is performed in two ways:

1. The participant's use of a website or application is recorded behind the scenes and then the data are transmitted to the test team (Hong & Landay, 2001). A long list of tools for remote recording can be found at http://www.uu.se/Software/Analyzers/Access-analyzers.html
2. The participant works with two browsers, one with the product or prototype in it, the second with the instructions for the participant to follow. The instructions browser includes the tasks to be attempted, buttons to click at the beginning and end of a task, a free-form comment area, and questions or ratings to answer during or after each task.

The primary advantage of asynchronous over synchronous testing is a larger sample size, because the number of participants is not limited by time requirements of the administrator. Tullis, Fleischman, McNulty, Cianchette, and Bergel (2002) tested 88 participants in a short period. The disadvantage is that you cannot see or interact directly with each participant. In the study, the participants provided an unexpectedly large volume of feedback in the free form comment field, which provided insight into the usability problems with the product. Tullis et al. reported no substantial difference between asynchronous testing and laboratory testing in terms of performance measures and the number and types of problems identified.

While we have very little research to date on remote testing, its advantages make it likely to increase in popularity, and it may some day be as popular as lab testing is.

Onsite Testing

Onsite usability testing follows the same basic procedure as testing in a lab or remote testing. The primary advantage of onsite testing is that, like remote testing, participants work in their environments and do not have to travel to the lab. In addition, unlike most remote testing, you get to see the participants and their environments.

Onsite testing often does not require any special equipment. You can simply observe participants and manually record findings. There are portable labs to record the test session if needed. They can generally be set up onsite quickly.

There are several disadvantages of onsite testing (Mitropoulos-Rundus & Muszak, 1997). Onsite tests can be very expensive, as the test team has to travel to the participant. It also limits the number of team members who can view the test live. Too many

observers can be overwhelming to participants. The portable equipment can record the session, but as stated earlier, recordings are not as compelling as live tests. Finally, participants may be reluctant to allow testers into their offices or homes.

If the usability test is evaluating a website, the tester may have to make special arrangements. For example, if the prototype is kept on a development server inside your organization's firewall, the tester has to be sure there is a way for the participant to access it. Testers can bring in laptops running the software or website; however, the whole application must run on the laptop itself. You will (most likely) not be able to access the Internet through the participant's network. Further, if participants need accounts, testers should set this up ahead of time and make sure participants can access their accounts on the website.

Other usability methods are conducted in the participant's environment. The most well known is contextual interviewing. Ethnographic methods also help usability specialists understand the user's environment. These methods are outside the scope of this chapter, but they can all be useful. Usability specialists should consider the goals of their onsite work to determine which of these methods would be most appropriate.

In a recent study, cell phone usability was tested in both the lab and while the participants were making a trip using public transportation (Kaikkoner, Kekalainen, Canker, Kallio, & Kankainen, 2005). Each environment had 20 participants. The results showed only minor differences in the problems identified by each method. Kaikkoner et al. (2005) concluded that, for cell phone usability, the extra resources and time required for the field testing were not worth the payoff.

CORE ISSUES IN USABILITY TESTING

The Measurement of Usability

To sufficiently understand and address usability issues, it is important to consider the types of data collected and how to analyze the data. Until recently, usability professionals have generally agreed that quantitative measures of usability should include effectiveness, efficiency, and satisfaction (as described in ANSI, 2001). Beginning in about 2000, there has been more interest in additional measures of affect such as fun, challenge, and stimulation (e.g., Pagulayan, Steury, Fulton, & Romero, 2003).

Effectiveness refers to whether the user can complete tasks; it is evaluated with measures such as success rates, error rates, number of help desk calls, and number of assists.

Efficiency refers to the effort it takes to complete the tasks. Task time, a performance-based, quantitative measure, has been the primary measure of efficiency. Investigators can report times for the entire task or for each step towards the goal. For websites, additional measures of efficiency may be useful, such as the number of pages accessed compared to the minimum number required to complete a task (Smith, 1996).

One measure that has been proposed using time measures to evaluate efficiency is the Novice/Expert Ratio (Urokohara, Tanaka, Furuta, Honda, & Kurosu, 2000). This measure is the task time for a participant divided by the average time for some referent group, such as experts. When the average time is an ideal, this measure can provide some useful insight. However, the ratio has proven difficult to interpret (Sauro & Kindlund, 2005).

User satisfaction refers to the participants' opinions and attitudes about the product, and it is generally evaluated with satisfaction ratings, questionnaires, and participant comments. Questionnaires for measuring user satisfaction include the Questionnaire for User Interaction Satisfaction (QUIS; Chin, Diehl, & Norman, 1988), the Software Usability Scale (SUS; Brooke, 1996), and the Software Usability Measurement Inventory (SUMI; Kirakowski, 1996). Tullis and Stetson (2004) provided a good description and analysis of user satisfaction questionnaires in usability tests, and their study showed that these questionnaires are not reliable with fewer than 10–12 respondents, which makes their use in a diagnostic usability test problematic.

There is some debate about how strongly measures are correlated. Frøkjær, Hertzum, and Hornbæk (2000) found only a weak correlation, but Sauro and Kindlund (2005) found a stronger one. Further, measures of effectiveness and efficiency (generally objective performance measures) may have a low correlation with satisfaction measures, as participants often blame themselves for problems rather than the product. In some cases, effectiveness and efficiency measures may even conflict with each other (Nielsen & Levy, 1994). Therefore, it is important to evaluate all of these measures together to get a full understanding of a product's usability because each assesses a different aspect of usability.

In addition to these quantitative measures, usability tests can also provide useful qualitative data, including participant comments and investigator observations, which can provide profound insight into usability problems and solutions. The process of interpreting them is part of the art of usability testing. Some usability professionals believe that the qualitative results are actually more informative and useful than the quantitative results. For example, Ebling and John (2000) found that over half of the problems identified in their test came from the think aloud protocol alone.

Given the number of usability measures, it seems logical to try to simplify them into a single usability metric. This metric would be useful in determining when a product is usable enough or in comparing usability of several designs (it would probably be less useful in formative usability tests, where the qualitative data that yield design solutions are so critical). Sauro and Kindlund (2005) and McGee (2003, 2004) reported on options for a single metric they have used successfully. These metrics require a fair amount of attention to the details of data collection and analysis, but the authors felt the benefits are worth the additional work.

Assigning severity ratings. Several recent research studies have looked at the validity and the reliability of severity rating scales. Unfortunately, there is little consistency in the use of severity ratings. This lack of consistency appears in all forms of usability evaluation and is one of the most important challenges to usability methodology.

Most studies have looked at the inconsistencies among experts using severity scales with inspection methods such as heuristic evaluation. However, Jacobsen, Hertzum, and John

(1998) found the same results with usability testing. They asked four experienced usability testers to watch tapes of the same usability test and then identify problems, including the top 10 problems in terms of severity. Of the 93 problems identified with the product, only 20% were detected by all evaluators, while 46% were only found by a single evaluator. None of the top 10 severe problems appeared on all four evaluators' lists. Molich and Dumas (in press) found that 25% of the problems reported in common by two or more evaluation teams were classified into different severity categories.

Lesaigle and Biers (2000) reported a disappointing correlation coefficient (+0.16) among professional testers' ratings of the severity of the same usability problems in a usability test. They used Nielsen's severity rating scale (Nielsen, 1992). Cantani and Biers (1998) found that severity ratings by usability professionals were not consistent.

The results of these studies strike a blow at one of the most often mentioned strengths of usability testing—its ability to uncover the most severe usability problems. Now, we do not know whether the inconsistencies in severity judgments are the result of the poor scales, the differing perceptions of usability specialists, or both.

Several practitioners have proposed severity rating schemes: Dumas and Redish (1999), Nielsen (1992), Rubin (1994), and Wilson and Coyne (2001). The schemes have three properties in common:

1. They all start with a rating scale that is derived from software bug reporting. The most severe category usually involves loss of data or task failure, and the least severe category involves problems that are so unimportant that they do not need an immediate fix. The middle levels between the extremes are usually difficult to interpret and they are stated in words that are hard to apply to specific cases. For example, Dumas and Redish (1999) proposed two middle levels: (a) problems that create significant delay and frustration and (b) problems that have a minor effect on usability. Nielson's (1992) middle levels are (a) major usability problem—important to fix so it should be given high priority and (b) minor usability problem—fixing is given low priority. None of the sources gives practitioners any guidance on how problems fit into these scale levels, especially the middle ones.
2. All of the authors admit, at least indirectly, that their scales alone are not enough to assess severity. The tester must consider one or more other factors in judging severity. For example, Nielsen (1992) described four factors in addition to the severity rating: frequency, impact, persistence, and something called "market impact." Rubin (1994) proposed to multiply the rating by the number of future users who will have the problem. Dumas and Redish (1999) added a second dimension—the scope of the problem either local or global, with no levels in between. With the exception of Rubin's multiplication rule, none of these other factors is described in enough detail to indicate how their combination with the severity scale would work, which is, perhaps, an indicator of the weakness of the severity scales themselves.

None of the scales indicates how to treat individual differences. For example, what does one do if only two of eight participants fail to complete a task because of a usability problem? Is that problem placed in the most severe category or does it move down a level? If a problem is global rather than local, does that change its severity? The authors of these scales provided little guidance.

In summary, severity ratings can be helpful in determining which problems to fix. However, the methods of assigning severity are complex and not yet fully understood.

Eye tracking. Eye tracking has been slow to gain advocates in usability testing (e.g., Bojko & Stephenson, 2005; Goldberg, Stimson, Lewenstein, Scott, & Wichansky, 2002; Russell, 2005). Eye movements can be very helpful in identifying exact sources of confusion or errors. For example, if users do not follow instructions on the screen properly, it could be either because they do not understand them (as evidenced by long gazes at the text) or because they do not see the target (as evidenced by no glances at the text). Clearly, eye tracking data can help designers identify the causes of usability problems and recommend appropriate solutions.

The eye tracking technology has advanced recently. Participants are no longer required to wear awkward and bulky devices. In fact, the cameras used to track eye movement are noninvasive, and are almost unnoticeable to participants. Further, the software available to analyze the data has also improved. This advancement is critical, as eye trackers can collect as many as 60 data points per second. However, eye tracking still has some disadvantages. For example, some participants must calibrate the system several times before they are successful, which increases their burden. Some participants (e.g., those who wear bifocals) cannot be tracked at all. Further, eye tracking cannot be used with think aloud methods because studies have shown the method affects where participants look on the display (Bojko, 2005). Eye tracking is useful for selected tasks during which it is important to understand the details of what participants are looking at (Bojko & Stephenson, 2005)

The Complexity and Importance of Thinking Aloud

One of the early differences between a usability test and a research study was that the test participants typically thought aloud in a usability test. While concurrent thinking aloud is normally done as part of a diagnostic usability test, it is really a method of its own. It has been used in conjunction with many other methods, such as the evaluation of the usability of print documentation (Soderston, 1985).

Thinking aloud has been used in psychological research since the turn of the 20th century, but it is best known as a cognitive psychology method for studying short-term memory (Ericsson & Simon, 1993).

Thinking aloud provides usability testing with most of its drama. Without thinking aloud, it is unlikely that usability testing would have become the most influential usability engineering method. The think aloud protocol grabs the attention of first-time visitors to usability test and gives a test session the appearance of a science-based method.

When usability testing was first being codified, thinking aloud was borrowed from cognitive psychology without much

reflection. It was not until shortly after 2000 that usability specialists began to look at it more closely. Independently, Boren and Ramey (2000) and Dumas (2001a) looked more closely at what Ericsson and Simon (1993) described as the think aloud method used in cognitive psychology research and whether testing practitioners were really following that method. Both reviews showed that the descriptions of how to use the think aloud method that had been provided to usability testing practitioners by Dumas and Redish (1999) and Rubin (1994) were in direct contradiction to the instructions used in cognitive psychology research. In cognitive psychology research studies, participants are discouraged from reporting feelings or expectations or from making any verbal diversions over and above the content of their actions. In usability testing, participants are encouraged to report on their feelings and expectations and on additional relevant issues.

Only a few research studies have been done on the think aloud method in a usability-testing context. Krahmer and Ummelen (2004) compared typical usability testing think aloud instructions to the instructions used by Ericsson and Simon (1993) and found that the research instructions do not work well in a testing context. Ebling and John (2000) traced each usability problem found in a usability test back to its source in the test measures. They found that over half of the problems identified in their test came from the think aloud protocol alone. Their study supplemented an earlier one by Virzi, Source, and Herbert (1993), which showed that fewer problems are identified when the participants do not think aloud.

One of the interesting questions about thinking aloud is whether it influences what test participants do in a usability test session. Bojko (2005) reported that thinking aloud changes the way participants examine a web page and recommended that thinking aloud not be used with eye tracking. That recommendation raises interesting questions about whether thinking aloud adds to the artificiality of the testing situation. Further, as stated earlier, Boren and Ramey (2000) found that test administrators use the think aloud method differently. Therefore, it is important to plan carefully how thinking aloud will be incorporated into a usability test.

For the practitioner, this research confirms the importance of thinking aloud as both a method to catch the eye and ear of visitors and to uncover usability problems. Most moderators have not changed their approach to think aloud instructions since the early 1990s. However, we are just beginning to understand the thinking aloud process. We do not have basic information such as whether the presence of a moderator in the room increases or decreases thinking aloud verbalizations or whether thinking aloud changes the way participants examine a product.

How Many Participants Are Enough?

Part of the popularity of usability testing comes from its ability to find usability problems with only a few participants. Anyone who watches multiple test sessions with the same set of tasks perceives that the same issues begin to repeat and that somewhere in the five to eight test participant range, with the same user population, it begins to seem unproductive to test more participants. So it was with great joy that testers greeted the research studies by Virzi (1990, 1992) showing that 80% of the total number of usability problems that will be uncovered by as many as 20 participants will be found by as few as 5. Virzi also found that those five participants uncover most of the problems judged as severe. This finding has been confirmed several times (Faulkner, 2003; Law & Vanderheiden, 2000). Practitioners continue to test with small numbers of participants, confident that they are finding most of the problems.

There have been a few challenges to the generality of this finding, most notably by Lewis (1994, 2001). The challenges make the reasonable case that not all tests are the same. A moderately complicated product being tested for the first time might indeed yield many of its problems to five to eight participants. But what about a product that is being retested after most of its problems have been fixed? Or a very large and complicated product? In each of these cases, one might expect that it might take more participants before problems begin to repeat. Lewis (1994) tested a large product, a suite of office productivity tools, with six-hour sessions and did not see the same convergence as Virzi (1990, 1992) had. It may also take more participants if the user population is very heterogeneous, such as with elderly and disabled users (Grossnickle, 2004; ITTAC, 2004; Swierenga & Guy, 2003). Lewis (2001) created a formula for determining how many participants are needed in a variety of testing situations.

The CUE studies (discussed in more detail later) show that there are hundreds of usability problems with any moderately complex product, but that any one test will only uncover a small fraction of them. However, Molich and Dumas (in press) argued that having a test team run more participants is not the way to find more problems. Some limitation causes one test team with one sample of tasks to find only a fraction of the available problems. We do not know yet why this happens.

Therefore, the answer to the question, "Is five enough?" is "It depends." It is yes in the sense that one team with one sample of tasks is unlikely to find more problems that it would consider severe by running more participants. It is no in the sense that there may be many more problems left to be uncovered.

Allowing Free Exploration

An important issue in user testing is what the participant does first in a test session. For example, if all users will have some training before they use the product, testers might want to be sure all participants get the same training (perhaps by providing it themselves).

There is often a preamble to the first task scenario that puts the test and the tasks into some context. Most often, the preamble leads to the first task scenario. Some testers argue that this procedure is unrealistic and that in the real world people do not work that way—they spend a few minutes exploring the product before they start doing tasks. Others argue that going directly to tasks without training or much of a preamble puts stress on the product to stand on its own—stress that is useful for making products more usable.

Should testers consider allowing the test participants 5–10 minutes of exploration before they begin the task scenarios?

Those in favor of free exploration argue:

- Without exploration, the product is getting an unfairly hard evaluation.
- Without exploration, the testing situation is not simulating the real use environment, especially for web-based products. Users must know something about the product to buy it or their company might give them some orientation to it.
- In some special situations, such as with a new game, testers gain valuable information by allowing new users to explore (Pagulayan, Keeker, et al., 2003).

Those against free exploration argue,

- Free exploration introduces added variability into the test; some participants will find information that helps them do the tasks, while others will not find that information.
- The test is intended to be a hard evaluation for the product to pass.

This debate continues, but most testers do not allow free exploration.

Reliability of Usability Testing

Beginning in 1998, a series of experiments and analysis papers questioned the reliability of assessment methods in general and usability testing in particular. Reliability refers to whether usability testing will repeatedly find the same problems, given the same product, tasks, users, and environment. Almost from its beginning, usability testing was assumed to find most problems with even a single test. In fact, many researchers proposed that testing was a kind of gold standard against which other assessment methods were to be measured.

Desurvire, Kondziela, and Atwood (1992) conducted the first gold standard study. They were interested in how well an inspection predicted the results from a usability test. Their assumption was that usability testing finds the true problems. The three experts they used found less than half of the problems that were uncovered by Desurvire et al. in their usability test. Furthermore, the experts found more minor usability problems, leading Desurvire (1994), later interpreting her data, to say, "The training and experience of human factors experts seem to sensitize them so that they are better at predicting a user's minor problems with a system" (p. 186).

The use of testing data as a benchmark against which to compare other evaluation methods continues to the present day (Andre, Hartson, & Williges, 2003; Bailey, Allan, & Raiello, 1992; Sears, 1997). In these studies, any usability issue found by an inspection method is considered a false alarm if it is not also found by usability testing.

Jacobsen, Hertzum, and John (1998) were the first to study reliability in usability testing. They studied how evaluators differ when analyzing the same usability test sessions. In the study, four usability testers independently analyzed the same set of videotapes of four usability test sessions. Each session involved a user thinking aloud while solving tasks in a multimedia authoring system. As many as 46% of the problems were uniquely reported by single evaluators and 20% by only two evaluators.

A series of studies pursued this issue of reliability of testing further (Molich, Bevan, et al., 1998; Molich, Meghan, Ede & Karyukin, 2004; Molich & Dumas, in press). Known as the "comparative user evaluation" (CUE) studies, these papers reported studies in which multiple independent usability tests of the same product are done by professional usability teams. The CUE-1 and CUE-2 studies (Molich, Bevan, et al., 1998; Molich, Meghan, et al., 2004) documented wide differences in reported problems, methods, tasks, and usability reports. For example, in the CUE-2 study, nine teams independently usability tested the same website. Only a single team reported 75% of the 310 usability problems. At least 29 serious or critical problems were reported by single teams. CUE-2 demonstrated that the effectiveness of a usability test could depend on the tasks, the methodology, and the persons in charge of the test. Not all usability tests and testers are equal—even amongst professional organizations.

In the CUE-4 study, 17 teams conducted either an inspection (eight teams) or a usability test (nine teams) of the same hotel reservations site (Molich & Dumas, in press). These teams were chosen because most were well-known, published usability professionals. Again, the overlap among the usability problems was disappointing. None of the serious usability problems was found by all 17 teams or all 9 of the usability test teams. The average percentage overlap between pairs of teams that conducted usability testing was only 11.5%. Two of the testing teams started with the same task set, though one of them did modify some tasks. Those two teams did find more problems in common, 24%, but not as many as would be expected if dissimilar tasks were the main cause of the unreliability.

Hertzum and Jacobsen (2001) did an analysis of comparative studies of testing and inspections to date and termed the lack of reliability as the "evaluator effect." They recommended multiple independent evaluations of products, regardless of the assessment method used.

What this means for the practitioner. Despite these troubling findings, they have not impacted usability-testing methodologies significantly. The usability community is aware of many of the pitfalls of usability tests, but the research has not yet completely uncovered how to alleviate these problems. There is some evidence that the tasks selected affect the outcome. In addition, reliability is reduced when the usability testers do not interact with the development team. The CUE studies show the importance of considering tasks carefully and working with the development team to ensure reliable results. Further, the CUE studies reveal that one usability test is not sufficient to identify all usability problems with a product. A single usability test should be part of a larger effort involving multiple usability tests (ideally with independent testers) as well as other usability methods. The usability community needs to place a high priority on identifying the causes of this evaluator effect and ways to alleviate them.

There is an interesting connection between the evaluator effect and the five is enough issue we previously discussed. If the evaluator effect proves to have wide generality, it still does not mean that running more participants in a test will uncover more problems. It still appears that with one test team and one set of

tasks only so many problems will be uncovered. Running more participants under the same conditions will not find more issues in most cases; it will just use more resources. Conducting additional independent tests with different teams will find more problems but there is not yet enough empirical evidence to convince organizations to do so.

Validity of Usability Testing

While we have discussed the research and analysis about the reliability of testing, there has been almost nothing said about its validity. Validity always has to do with whether a method does what it is supposed to do. Perhaps the validity of usability testing has been ignored because of its robustness. Testing always finds strengths and weaknesses in a product. Furthermore, as we described previously, the Virzi (1990, 1992) studies provided evidence that problems judged as severe were found early in testing. There has never been a study published questioning whether testing finds problems. Is finding some of the problems enough?

To assess the validity of usability testing, we must first agree on what a usability test is supposed to do. Prior to the mid-1990s, the usability community used diagnostic tests primarily to uncover usability problems. The more problems they found, the better and, of course, the tests should find the most severe ones. Because testing has never been viewed as the only usability evaluation method to apply during development and because, ideally, there are iterative tests performed, it was not essential or expected that one test would find all of the problems. Still, the Virzi (1990, 1992) studies, and those that have confirmed his basic findings, indicated that most of the problems that one team with one set of tasks will uncover are found early and that 5–10 participants would find almost all of the problems that a team is likely to find.

On the other hand, the CUE studies suggest that one test team will find only a small fraction of the total problems and that, perhaps, finding problems should only be a partial indicator of an effective test. The RITE method, discussed below, suggests two additional possibilities for goals:

1. A test should result in an improved product design that fixes as many problems in the original design as the testing resources allow.
2. A test should increase the commitment of development teams to user-centered design and their willingness to pursue it for future projects.

Related to this second goal is the necessity for positive feedback on the effective aspects of a design. If one of the goals of testing is team building, then emphasizing the positive needs to be one of the main components of a test. Apprentices learning to conduct tests need to be taught the skill of noting what is working and making it part of the discussion with developers and test reporting.

Until we sort out the importance of these goals (finding problems, creating an improved design, and team building), we cannot fully understand the validity of what is arguably our most powerful evaluation tool.

Testing With Special Populations

As manufacturers push to expand their market share, they look to gain users in populations they were not serving before. Designing for these populations often requires special attention to the usability techniques used. This section addresses some of the special considerations for usability tests with international users, disabled users, elderly users, and children.

International Participants

Many manufacturers look for new customers across the globe. However, preparing a product for a new market may involve more than simply translating the language. Cultural differences can also impact appropriate design decisions such as color selections and the use of images. These differences can also impact the appropriate structure for web applications. Because of the significant differences across cultures, this it is important to conduct usability testing with participants from all the target cultures.

International usability testing follows the principles and theories of generic usability testing. However, a variety of challenges with testing participants in other cultures generally does not apply when testing in one's own culture. The challenges of communication and cultural differences are described in the following sections.

Communication. One of the most significant challenges with international usability tests is communication. Often, there are different languages. Sometimes the testers are bilingual, but often the tester must have help recruiting participants, preparing test materials, conducting the test, analyzing the results, and writing the report. Nielsen (1996) and Vatrapu and Pérez-Quinones (2004) offered several suggestions, including the following:

- Use employees of the company who live and work in that country. This may require training the employees to facilitate a usability test.
- Conduct the test in the participant's language using an interpreter.
- Hire a local usability firm.
- Run the test remotely.
- As a last resort, conduct the test yourself in your language, though this method is likely to be unnatural for the participant.

Tests that are conducted in the participant's language must be translated. Some testers prefer to have the translator work real time during the test. The translator can either serve as a liaison between the tester and the participant (adding significant time to the test) or between the test administrator and participant (who are both speaking the same language) and the observers. The tester may also have to arrange to provide the test report in more than one language.

Cultural differences. Other cultural differences may also impact a usability test. For example, gender expectations may affect who would facilitate a test. There may be gestures

considered natural or friendly in one culture, but offensive in another. Loring (2002) reported how understanding Hofstede's (1997) cultural dimensions for your target audience can help improve the user-centered design process and the final product. Vatrapu and Pérez-Quinones (2004) reported that when both the participant and the test administrator were from the same culture, the participants engaged in more think aloud behavior and the usability tests revealed more problems.

Disabled Participants

Usability tests with disabled participants require careful planning. Testers must understand the participants' disabilities and adjust their procedures accordingly. Several researchers published lessons learned from their experiences with disabled participants (Coyne, 2005; Grossnickle, 2004; ITTAC, 2004; Lepistö & Ovaska, 2004; Swierenga & Guy, 2003). Some of these lessons include:

- Recruiting disabled participants is more time consuming than recruiting general population participants. Local organizations and support groups may be willing to help.
- Disabled participants may need assistance getting to the usability lab.
- Consent forms must be accessible to all participants.
- Blind participants may require electronic or Braille versions.
- Participants with learning or cognitive disabilities may require special consideration to ensure they understand the test and their rights.
- Deaf participants may require a sign language interpreter who needs to be informed about the goals of the study.
- Participants with cognitive disabilities may require extra assistance understanding the tasks and may have trouble thinking aloud.
- Participants with physical disabilities may require adaptive technology to interact with the computer. Be sure the devices are working before participants arrive.
- Because of the great variability in disabilities, it may take more participants than typical usability tests.
- It can be especially difficult to observe participants who use Braille readers, as there is currently no good way to follow what the participant is reading.

Overall, tests with disabled participants may take longer than expected; testers should schedule enough time so participants are not rushed. Further, participation may be more taxing than for general population users, so the test should limit the number of tasks evaluated (Coyne, 2005). Finally, testers should ask participants before the test whether they need any special accommodations.

Elderly Participants

As the population ages, manufacturers are looking to expand their market to this growing population. Seniors are more active than ever. As a result, many manufacturers are working to ensure that their products are usable by their older users.

As people age, the diversity in their abilities increases. They may also have disabilities, such as those mentioned in the previous section. Many of the concerns and issues mentioned above also apply with elderly participants. In general, testers should be prepared for each participant, leaving plenty of time for each person.

There may also be generational issues. Testers should be aware of what their participants expect regarding social interaction and courtesy. Chisnell, Lee, and Redish (2005), Coyne (2005), and Tedesco, McNulty, and Tullis (2005) provided some guidance based on their experiences running usability test with older participants.

Children as Participants

When designing a product for children, usability tests must target children. Although the process is generally the same as with adult participants, there are a few important differences.

Recruiting children actually involves recruiting their parents. Patel and Paulsen (2002) suggested several good sources for recruiting. They recommended building rapport with organization leaders and parents. It is important to pay attention to the needs of both the parents and the child. Sometimes it is necessary to have the parents in the room during the test, especially for very young children. Investigators should be sure that the parents do not unnecessarily interfere with the test. However, investigators should be flexible, as each family will be different.

Investigators may want to alter the usability lab itself to be a better environment for children. Most usability labs use a standard office layout and décor. Although this is fine for testing adults, it is not the most welcoming to children. Making the room more child friendly can make children more comfortable and willing to participate.

The tasks should accommodate the abilities of children in the target age group. Investigators should consider (a) the wording of the instructions to be sure they are at an appropriate grade level and (b) whether the participants are old enough to complete the tasks. For example, preschoolers may not be able to perform a task and think aloud simultaneously. Also, because children often have short attention spans, the entire test session should be kept to a minimum.

Finally, what motivates adults does not always motivate children. Hanna, Risden, and Alexander (1997) suggested age-appropriate approaches for motivating young participants to continue. Most likely, the best approach for a preschooler is very different from that for a teenager.

Children can be unpredictable, so one or more members of the test team must understand the skills, abilities, and expectations of the children in the target user population. This will help testers to respond appropriately to unexpected situations.

Testing With Special Products

Usability methods tend to be consistent across products, applications, and systems. However, there are cases where there are issues specific to the type of product. This section addresses some of these issues and what researchers have found.

Mobile devices. In most cases, it is easy enough to observe the participants interact with the product being tested. For software, there are tools to capture the screen or stream the participant's display to another monitor. For hardware, video cameras can record the user's interaction. However, mobile devices, such as cell phones and PDAs, generally cannot be displayed on a computer monitor, and it is difficult to capture their interaction on video. Because they are mobile devices, users are often moving while using them.

Most mobile devices do not have a video out feature (Catani, 2003). They must be viewed using a camera. Catani (2003) described several methods of videotaping the users' interaction with mobile devices. The goal is to keep the users' experience as realistic as possible, while still allowing a good recording of the device. Either the device must be kept immobile so the camera can capture everything, or the camera must track the device. Catani provided several alternatives, giving the benefits and drawbacks of each.

Documentation. Ideally, all components of a system or product would be subject to usability testing to insure they are all well designed. This is the idea behind the goal of designing the user experience. A person's experience with an individual product goes beyond simply the product itself. It includes the shopping experience, customer services, and technical support. It also includes all documentation—installation guides, users' manuals, online help, and so forth.

Ensuring that the documentation is usable can be especially challenging. To evaluate documentation, there should be a reasonably functional prototype and reasonably complete documentation. Unfortunately, this does not usually happen until so close to deployment that it is difficult to make many changes.

Further, people generally try to avoid using manuals and help. They might even turn to colleagues before looking at a manual. As a result, it is difficult to generate scenarios in which the participants would naturally use the documentation. Skelton (1992) suggested the following ways to test documentation:

- Ask the participant to complete a task according to the documentation.
- Encourage the participant to review the documentation as they use the product.
- Ask the participant to find information in the documentation.
- Ask the participant to paraphrase or summarize what they read.

Measuring the usability of documentation is also a challenge. We often measure the usability of a product by how many times participants consult the manual or online help. However, if a participant still cannot complete a task after consulting the manual, it may be difficult to distinguish problems with the product from problems with the manual. A few measures may provide some insight, such as:

- Time spent reading the manual
- Number of times the participant looked at the table of contents
- Number of times the participant looked at the index
- Number of pages viewed.

These should be considered within the context of the whole study to identify problems. For example, users may need to view a variety of pages for a particular task, so what seems like a high number may actually be reasonable.

The Ethics of Usability Testing

In addition to their general ethical responsibilities as professionals, usability testers and organizations that conduct tests have special ethical requirements. There are three major areas of concern:

1. Informed consent
2. Confidentiality of data
3. Balancing harm with purpose

The principles of ethics have been extensively addressed by all of the relevant professional organizations:

- The American Psychological Association's code was revised in 2003 (available at http://www.upassoc.org/about_upa/structure_and_org_chart/code_of_conduct.html)
- The Usability Professionals' Association's code was updated in 2005 (available at http://www.apa.org/ethics/code2002.html)
- The Human Factors and Ergonomics Society's code was revised in 1998 (available at http://www.hfes.org/web/AboutHFES/ethics.html)

Informed consent. This topic has had the most written about it. Every testing organization has the responsibility to have an adequate informed consent form and every test session should start with both an explanation of the form and adequate time for the participants to read and sign it. The form must address the rights of the participant, especially the right to withdraw without penalty, any risks involved, and how the data, especially audio and video recording, will be handled. Any test session conducted without such consent is unethical and any organization or consultant who conducts testing without a proper form violates the above codes. A useful discussion about creating an effective consent form can be found in Waters, Carswell, Stephens, and Selwitz (2001).

Confidentiality of data. It is the responsibility of the test administrator to safeguard the identities of test participants and the mapping of their identities to data and recordings. Participants' names should not be used on forms or labels and recordings must be restricted to the purpose described in the consent form. Testers need to be especially careful about the use and distribution of highlight tapes.

Of special concern is the use of data or recordings when the participants are employees of the organization developing the product being tested. When supervisors or managers of participants attend a test session, the moderator has a special responsibility to inform the participants before they come to the session so that participants are not faced with the pressure of deciding whether to participate while knowing that they are being observed by supervisors or managers. In tests such as these,

it also is advisable to avoid the creation of highlight tapes or video clips that accompany test reports.

Balancing harm with purpose. Researchers must ensure the purpose of research activities justifies the risks to which participants are exposed. As we have discussed in this chapter, most usability tests are not considered research and very few testing situations involve any physical risk. However, to a layperson, testing looks just like research, and we know that there is emotional stress involved in most test sessions.

How can we justify putting participants under such stress and remain true to the ethical principles of our professions? Moderators typically tell participants before the tasks are attempted that they are not being tested, the product is. However, we know that most participants still blame themselves when they struggle or fail. Furthermore, moderators sometimes ask participants to continue even when participants say they cannot make further progress.

We believe that the justification for putting participants under stress comes from the balancing principle that future users are being saved stress by finding its causes in the test. Having a few participants struggle can provide the motivation for developers to fix problems, thereby preventing those problems for many future users. It is sad but true that sometimes development teams need to see participants fail repeatedly before they will expend resources on implementing solutions.

Test moderators, therefore, walk a tightrope between allowing participants to struggle and minimizing stress. Good moderators look for signs of emotional distress and use their judgment about how long to continue with tasks or the session. The usability community has not adequately addressed this ethical dilemma, and therefore, it demands immediate attention. Moderators need more guidance and honest discussion about how to make these important decisions.

Usability Testing to Show the Value of Usability

One of the important assets of testing is that it sells itself. Watching even a few minutes of live testing can be very persuasive. There are two reasons why testers need to get key project staff and decision makers to come to watch a test session:

- When people see their first live test session, they are usually fascinated by what they see and they understand the value of the method better. Watching a videotape of a session does not provide the same experience. It is important to expend whatever effort it takes to get these people to attend test sessions.
- When developers see live sessions, it is much easier to communicate the results to them. When they have seen some of the usability problems themselves, they are much less likely to resist agreeing on what the most important problems are. Some of them will even become advocates for testing.

Even though testing is known and accepted by a much wider circle of people than it was 10 years ago, the experience of watching a user work at a task while thinking aloud still converts more people to accept user-centered design practices than any other development tool.

FUTURE DIRECTIONS IN USABILITY TESTING

In this final section, we explore new directions in what has been, over the past 15 years, a very stable and robust method for the empirical evaluation of usability. As would be expected with new directions, it is not always clear where they will lead. However, we believe that the three areas we discuss have the potential to redirect the way we test and expand the scope of our test objectives.

Objectives Beyond Productivity

Since the year 2000, there has been an expansion of the scope of usability. The expansion has come from two directions:

- An emphasis on the qualities of a product design beyond usability and productivity to qualities such as fun and pleasure
- The recent evolution from user-centered design to user experience design, in which the usability of a product is necessary but not sufficient for product success

In both of these movements, the scope of evaluation expands. The question for usability testers is whether these movements require only that additional measures be added to a traditional test or whether these movements can be accommodated within usability testing at all.

There has been a growing sense that the more traditional definitions of usability are limiting, especially as we have sought to integrate usability assessment into earlier stages of design, and to cost justify our practices. These efforts have led us to realize that task effectiveness, efficiency, and satisfaction are only part of the story; affective aspects of the user's interaction are also critically important to ensuring the success of a product. The result has been an expansion of what usability professionals focus on, and a clouding of what is meant by usability and usability assessment. For example, Jordan (2002) and Hancock, Pepe, and Murphy (2005) drew upon Maslow's (1970) hierarchy of needs to create a hierarchy for consumer product needs. Here, functionality is essential, usability is important (and now expected), and pleasurable interactions are what users really want. Hancock, Keeker,, et al. called this concept "hedonomics."

Some of the most interesting concepts about dimensions beyond usability have come from the development of computer-based games. According to Pagulayan, Keeker, et al. (2003), games differ from productivity applications in several important ways:

- While productivity applications are tools to accomplish work, games are intended to be a pleasure to play, though they can also teach and challenge.
- Games define their own goals rather than having their goals defined by the external environment.
- Each game needs to be noticeably different from its competitors and capable of grabbing the users' attention.
- Rather then stressing consistency, game design stresses variety and innovation.

- Games deliberately impose constraints rather than remove them.
- Successful games avoid simplicity, presenting a challenge to users without being so complex that most users lose most of the time.

Pagulayan, Steury, et al. (2003) described the methods they use to evaluate games, including the RITE method described below.

One of the additions to the toolkit of usability testing is the use of scales for measuring product qualities such as pleasure and joy. Hassenzahl, Beu and Burmester (2001) and Jordan (2000) described scales to measure the affect a design evokes. A related, but nonverbal, measure is the use of "emocards," which each depict drawn facial expressions of emotions (Desmet, 2002). Participants pick or rate the cards that best describe the feeling they get from a design.

It is unclear whether these scales are reliable with the small samples typical of usability testing, but it is clear that product designers do want to evaluate measures beyond traditional usability.

Asynchronous Testing

Previously, we discussed asynchronous testing as a variation of remote testing. Its major advantage is the large sample sizes that it can provide, which could be very important in an environment in which product qualities other than usability need to be assessed. The robustness of traditional synchronous testing at uncovering usability issues with very small samples may prove to be a major weakness if larger samples are needed to assess other product qualities. Furthermore, asynchronous testing could be combined with measures that have been used to study long-term ease of use issues rather than just initial ease of leaning issues. For example, methods such as user diaries and product reaction checklists (Jordan, 2000) could be used as part of an asynchronous test to gather changes to attitudes and usage patterns over time.

The RITE Method

A group of usability specialists at Microsoft created a new method that may look similar to traditional usability testing but is, in fact, quite different (Medlock, Wixon, McGee, & Welsh, 2005; Medlock, Wixon, Terrano, Romero, & Fulton, 2002). Known as the "rapid iterative test and evaluation" (RITE) method, it focused on fixing usability problems rather than just finding them. In outline, the method consists of the following:

- A team consists of the key decision makers for a product work together. The team includes the usability specialists responsible for the product, but also includes programmers/designers who can make changes to the product, as they are determined and, most importantly, someone in authority who can authorize changes.
- The team selects the tasks to be run and attends all sessions. As with traditional usability testing, testers recruit participants from the target market and use the think aloud method. Standard measures such as task failure and errors are recorded.
- After each session, the usability specialist identifies the problems and their severity. The team then decides if there is a fix for each problem, whether they can implement it, or whether they need to run more participants to be sure.
- The programmers/designers make agreed upon changes and the new version is run new participants.
- Again, problems are identified, including whether the fixes to previous problems have worked. If not, new fixes are created.
- The team decides which problems they can fix, which they need to examine in more detail, or which require resources are not currently available.
- Testers run additional participants until resources are expended or the team is certain that the majority of the problems it has addressed are fixed. Running these extra participants is a check that new problems have not been introduced by the design modifications.

This method requires more test participants than traditional usability testing and the number of participants required is unknown when the test is planned. It is reported that the total time from start to finished design is shorter than with traditional testing in which the solutions are not addressed until after the test sessions are over. Perhaps the most important contribution of this method is the team building it provides. Development teams are reported to like this method very much, and there are fewer disputes about the reality of problems and about why they need to be fixed. Furthermore, the immediate feedback the sessions provide motivates key stakeholders to attend sessions and stay involved in the process.

While there are some products or organizations that do not lend themselves to the use of this method, it holds great promise for making usability testing a positive experience for development teams and it appears to make it more likely that products will be improved (Medlock, Wixon, McGee, et al., 2005).

We believe that the RITE method is not just a variation of usability testing, but is in fact, quite a different method. It may be a misnomer to call it a usability test method rather than a design and evaluation method. The traditional usability test is based on the psychology research experiment. It is an evaluation method focused on identifying problems and, as we have described above, its validity is difficult to establish.

While the reliability of the RITE method is yet unknown, it is based on a different cause-effect model from the research experiment. It is closer to the model developed by Skinner (1956) in which an independent variable, such as a reinforcement schedule, is turned off and on successively to see how it affects performance. In a RITE test, the user interface is successively exposed participant by participant until errors and task failures are extinguished (that is, the product achieves the desired performance). When the test is finished, the quality of the test is assessed by its ability to have fixed the usability problems or, in some cases, to have identified a fix that will be implemented later. Consequently, its validity will be easier to establish.

CONCLUSION

Usability testing and usability labs established themselves very rapidly beginning about 1990, and they have remained remarkably stable since. An outside observer attending a typical test session in 2005 would see only subtle differences in the equipment and procedures from one conducted in 1990. However, the winds of change are in the air. Testing is moving out of the laboratory and timely testing with larger samples is now technically feasible. The push from user experience design and the interest in product pleasure are expanding testing's scope. Its ability to find the majority of problems with even moderately complex designs has been challenged, as has its narrow focus on problem identification. This has led to an increase in research regarding the characteristics of usability testing. We believe that usability testing will look very different in 2010.

References

Andre, T., Williges, R., & Hartson, H. (2003). The effectiveness of usability evaluation methods: Determining the appropriate criteria. *Proceedings of the Human Factors and Ergonomics Society, 43rd Annual Meeting* (pp. 1090–1094). Denver, CO.

American National Standards Institute. (2001). *Common industry format for usability test reports* (ANSI-NCITS 354-2001). Washington, DC: Author.

Bailey, R. W., Allan, R. W., & Raiello, P. (1992). Usability testing vs. heuristic evaluation: A head-to-head comparison. *Proceedings of the Human Factors and Ergonomics Society, 36th Annual Meeting* (pp. 409–413). Atlanta, GA.

Barker, R. T., & Biers, D. W. (1994). Software usability testing: Do user self-consciousness and the laboratory environment make any difference? *Proceedings of the Human Factors Society, 38th Annual Meeting* (pp. 1131–1134). Santa Monica, CA.

Bojko, A. (2005). Eye tracking in user experience testing: How to make the most of it. *Proceedings of the Usability Professionals' Association* (pp. 1–9). Montreal.

Bojko, A., & Stephenson, A. (2005). Supplementing conventional usability measures with eye movement data in evaluating visual search performance. *Proceedings of the 11th International Conference on Human-Computer Interaction* (*HCII 2005*) (pp. 117–121). Las Vegas, NV.

Boren, M., & Ramey, J., (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communication, 43*(3), 261–278.

Branaghan, R. (1997, April). Ten tips for selecting usability test participants. *Common Ground, 7*, 3–6.

Briggs, J., & Johnson, B. (2005). *Practical prototypes: High fidelity website prototypes in use.* Paper presented at the Usability Professionals' Association 14th Annual Conference, Montreal, CA.

Brooke, J. (1996). SUS: A quick and dirty usability scale. In P. Jordan, B. Thomas, B. Weerdmeester, & I. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). London: Taylor & Francis.

Catani, M. B. (2003). Observation methodologies for usability tests of handheld devices. *Proceedings of the Usability Professionals' Association 12th Annual Conference* (pp. 1–6). Scottsdale, AZ.

Catani, M. B., & Biers, D. W. (1998). Usability evaluation and prototype fidelity: Users and usability professionals. *Proceedings of the Human Factors Society, 42nd Annual Meeting* (pp. 1331–1335). Santa Monica, CA.

Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. *Proceedings of Human Factors in Computing Systems* (*CHI'88*) (pp. 213–218). Los Angeles, CA.

Chisnell, D., Lee, A., & Redish, J. (2005). *Recruiting and working with older participants.* Washington, DC: AARP. Retrieved October 13, 2005, from http://www.aarp.org/olderwiserwired/oww-features/Articles/a2004-03-03-recruiting-participants.html

Cockton, G., Lavery, D., & Woolrych, A. (2003). Inspection-based evaluations. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 1118–1125). Mahwah, NJ: Lawrence Erlbaum Associates.

Coyne, K. P. (2005). Conducting simple usability studies with users with disabilities. *Proceedings of HCI International 2005* (pp. 890–893). Las Vegas, NV.

Datz-Kauffold, B., & Henry, S. L. (2000). Waving magic wands: Interaction techniques to improve usability testing. *Proceedings of the Usability Professionals' Association 9th Annual Conference* (pp. 14–18). Asheville, NC.

Desmet, P. (2002). *Designing emotion*. Unpublished doctoral thesis, Delft University of Technology, Netherlands.

Desurvire, H. W. (1994). Faster, cheaper! Are usability inspection methods as effective as empirical testing? In J. Nielsen & R. Mack (Eds.), *Usability inspection methods* (pp. 173–202). New York: John Wiley & Sons.

Desurvire, H. W., Kondziela J. M., & Atwood, M. E. (1992). What is gained and lost when using evaluation methods other than empirical testing. In A. Monk, D. Diaper, & M. D. Harrison (Eds.) *, People and computers VII* (pp. 89–102). Cambridge, MA: Cambridge University Press.

Dumas, J. (2001a). Usability testing methods: Think aloud protocols. In R. Branaghan (Ed.), *Design by people for people: Essays on usability* (pp. 119–130). Chicago, IL: Usability Professionals' Association.

Dumas, J., (2001b). Usability testing methods: How many test participants in a usability test are enough? In R. Branaghan (Ed.), *Design by people for people: Essays on usability* (pp. 173–182). Chicago, IL: Usability Professionals' Association.

Dumas, J, (2003, November). Usability evaluation from your desktop. *Association for Information Systems* (*AIS*) *SIGCHI Newsletter, 2*(2), 7–8.

Dumas, J., & Redish, G. (1993). *A practical guide to usability testing*. Norwood, NJ: Ablex Publishing Co.

Dumas, J., & Redish, G. (1999). *A practical guide to usability testing* (Rev. ed.). London: Intellect Books.

Ebling, M., & John, B. (2000). On the contributions of different empirical data in usability testing. *Proceedings of Designing Interactive Systems* (pp. 289–296.). Brooklyn, NY.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as Data*. Cambridge, MA: MIT Press.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers, 35*(3), 379–383.

Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000). Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? *Proceedings of Human Factors in Computing Systems* (*CHI 2000*) (pp. 345–352). Fort Lauderdale, FL.

Gaba, D. M. (1994). Human performance in dynamic medical domains. In M. S. Bogner (Ed.) , *Human error in medicine* (pp. 197–224). Hillsdale, NJ: Lawrence Erlbaum Associates.

Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., & Wichansky, A. M. (2002). Eye tracking in web search tasks: Design implications. *Proceedings of the Eye Tracking Research & Applications Symposium* (*ETRA*) (pp. 51–58). New Orleans, LA.

Grossnickle, M. M. (2004). *How many users with disabilities should you include when conducting a usability test for accessibility? Idea market.* Paper presented at The Usability Professionals' Association Annual Conference, Minneapolis, MN. Retrieved September 13, 2005, from http://www.upassoc.org/usability_resources/conference/2004/im_martinson.html

Hancock, P., Pepe, A., & Murphy, L. (2005). Hedonomics: The power of positive and pleasurable ergonomics. *Ergonomics in Design, 13*(1), 8–14.

Hackman, G. S., & Biers, D. W. (1992). Team usability testing: Are two heads better than one? *Proceedings of the Human Factors Society, 36th Annual Meeting* (pp. 1205–1209). Atlanta, GA.

Hanna, L., Risden, K., & Alexander, K. J. (1997, September/October). Guidelines for usability testing with children. *Interactions*, 9–14.

Hassenzahl, M., Beu, A., & Burmester, M. (2001). Engineering joy. *IEEE Software*, 70–76.

Hertzum, M., & Jacobsen, N. E. (2001). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction, 13*(4), 421–443.

Hofstede, G. (1997). *Cultures and organizations: Software of the mind.* New York: McGraw-Hill.

Hong, J., & Landay, J. (2001). WebQuilt: A framework for capturing and visualizing the web experience, *Proceedings of the 10th International Conference on World Wide Web* (pp. 717–724). Seattle, WA.

Information Technology Technical Assistance and Training Center, Georgia Institute of Technology. (2004). *Planning usability testing for accessibility*. Retrieved September 13, 2005, from http://www.ittatc.org/technical/access-ucd/ut_plan.php

Jacobsen, N., Hertzum, M., & John, B. E. (1998). The evaluator effect in usability studies: Problem detection and severity judgments. *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 1336–1340). Chicago, IL.

Jordan, P. (2002). The personalities of products. In W. Green & P. Jordan (Eds.), *Pleasure with products* (pp. 19–48). London: Taylor & Francis.

Kaikkaner, A., Kekalainen, A, Canker, M. Kalliot, T, & Kankainen, A. (2005). Usability testing of mobile applications: A comparison between laboratory and field studies. *Journal of Usability Studies, 1*, 4–16.

Kennedy, S. (1989). Using video in the BNR usability lab. *SIGCHI Bulletin, 21*, 92–95.

Kirakowski, J. (1996). The software usability measurement inventory (SUMI): Background and usage. In P. Jordan, B. Thomas, B. Weerdmeester, & I. McClelland (Eds.), *Usability evaluation in industry* (pp. 169–177). London: Taylor & Francis.

Krahmer, E., & Ummelen, N. (2004). Thinking about thinking aloud: A comparison of two verbal protocols for usability testing, *IEEE Transactions on Professional Communication, 47*(2), 105–117.

Law, C., & Vanderheiden, G. (2000). Reducing sample sizes when user testing with people who have and who are simulating disabilities: Experiences with blindness and public information kiosks. *Proceedings of the Human Factors Society, 44th Annual Meeting* (pp. 157–160). San Diego, CA.

Lepistö, A., & Ovaska, S. (2004). Usability evaluation involving participants with cognitive disabilities. *Proceedings of NordiCHI* (pp. 305–308). Tampere, Finland.

Lesaigle, E. M., & Biers, D. W. (2000). Effect of type of information on real-time usability evaluation: Implications for remote usability testing. *Proceedings of the IEA 2000/HFES 2000 Congress* (pp. 6, 585–588). San Diego, CA.

Lewis, J. (1994). Sample size for usability studies: Additional considerations. *Human Factors, 36*, 368–378.

Lewis, J. (2001). Evaluation of procedures of adjusting problem discovery rates estimates from small samples. *International Journal of Human Computer Interaction, 7*(1), 57–78.

Loring, B. (2002). Conducting bilingual user research. *Proceedings of the Usability Professionals' Association Annual Meeting* (pp. 1–5). Orlando, FL.

Maslow, A.H. (1970). *Motivation and personality* (2nd ed.). New York: Viking Press.

McGee, M. (2003). Usability magnitude estimation. *Proceedings of the Human Factors Society, 47th Annual Meeting* (pp. 691–695). Denver, CO.

McGee, M. (2004). Master usability scaling: Magnitude estimation and master scaling applied to usability measurement. *Proceedings of Human Factors in Computing Systems* (*CHI 2004*) (pp. 335–342). New Orleans, LA.

Medlock, M., Wixon, D., McGee, M., & Welsh, D. (2005). The rapid iterative test and evaluation method: Better products in less time. In R. Bias & D. Mayhew (Eds.), *Cost-justifying usability: An update for the information age* (pp. 489–517). New York: Morgan Kaufman Publishers.

Medlock, M. C., Wixon, D., Terrano, M., Romero, R., & Fulton, B. (2002). Using the RITE method to improve products: A definition and a case study. *Proceedings of the Usability Professionals' Association* (pp. 1–12). Orlando, FL.

Meister, D. (1999). *The history of human factors and ergonomics*. Mahwah, NJ: Lawrence Erlbaum Associates.

Mitropoulos-Rundus, D., & Muszak, J. (1997, January). How to design and conduct a consumer in-home usability test. *Common Ground, 7*, 10–12.

Molich, R, Meghan R. Ede, K., & Karyukin, B. (2004). Comparative usability evaluation. *Behaviour & Information Technology, 23*, 65–74.

Molich, R., Bevan, N., Curson, I., Butler, S., Kindlund, E., Miller, D., et al. (1998). Comparative evaluation of usability tests [Electronic version]. *Proceedings of Usability Professionals' Association,* Washington, DC.

Molich, R. & Dumas, J. (in press). Comparative usability evaluation (CUE-4), *Behaviour and Information Technology*.

Muñoz, R., Miller-Jacobs, H. H., Spool, J. M., & Verplank, B. (1992). In search of the ideal prototype. *Proceedings of Human Factors in Computing Systems* (*CHI'92*) (pp. 577–579). Monterey, CA.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proceedings of CHI'92* (pp. 373–380). Monterey, CA.

Nielsen, J. (1994). Usability laboratories [Special issue]. *Behavior and Information Technology*, 13.

Nielsen, J. (1996). *International usability testing.* Retrieved September 13, 2005, from http://www.useit.com/papers/international_usetest.html

Nielsen, J., & Levy, J. (1994). Measuring usability: Preference vs. performance. *Communications of the ACM, 37* , 66–76.

Pagulayan, R., Steury, K., Fulton, B., & Romero, R. (2003). Designing for fun: User-testing case studies. In M. Blythe, K. Overbeeke, A. Monk & P. Wright (Eds.), *Funology: From usability to enjoyment* (pp. 137–150). Norwell, MA: Khuwer Academic Publishers.

Pagulayan, R., Keeker, K., Wixon, D., Romero, R., & Fuller, T. (2003). User-centered design in games. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook* (pp. 883–906). Mahwah, NJ: Lawrence Erlbaum Associates.

Patel, M., & Paulsen, C. A. (2002). Strategies for recruiting children. *Proceedings of the Usability Professionals' Association Annual Meeting.* Orlando, FL.

Perkins, R. (2001). Remote usability evaluation over the Internet. In R. Branaghan (Ed.), *Design by people for people: Essays on usability* (pp. 153–162). Chicago, IL: Usability Professionals' Association.

Pruitt, J., & Adlin, T. (2005). *The persona lifecycle: Keeping people in mind throughout product design*. San Francisco, CA: Morgan Kaufman.

Roberts, T., & Moran, T. (1982). Evaluation of text editors. *Proceedings of Human Factors in Computer Systems* (pp. 136–141). Gaithersburg, VA.

Rosenbaum, S., Rohn, J., & Humburg, J. (2000). A toolkit for strategic usability: Results from workshops, panels, and surveys. *Proceedings of CHI 2000* (pp. 337–344). The Hague, Netherlands.

Rubin, J. (1994). *Handbook of usability testing.* New York: John Wiley & Sons.

Rudd, J., & Isensee, S. (1994). Twenty-two tips for a happier, healthier prototype. *Interactions*, 35–40.

Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *Interactions*, 76–85.

Russell, M. (2005). Using eye-tracking data to understand first impressions of a website. *Usability News, 7*(1). Retrieved June 23, 2005, from http://psychology.wichita.edu/surl/usabilitynews/71/eye_tracking.html

Sauro, J., & Kindlund, E. (2005). A method to standardize usability metrics into a single score. *Proceedings of Human Factors in Computing Systems* (*CHI 2005*) (pp. 401–409). Denver, CO.

Scholtz, J., & Bouchette, D. (1995, March). Usability testing and group-based software: Lessons from the field. *Common Ground, 5*, 1–11.

Sears, A. (1997). Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human Computer Interaction, 9*, 213–234.

Shneiderman, B. (1987). *Designing the user interface: Strategies for effective human computer interaction* (1st ed.). Reading, MA: Addison-Wesley.

Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human computer interaction* (2nd ed.). Reading, MA: Addison-Wesley.

Shneiderman, B. (1997). *Designing the user interface: Strategies for effective human computer interaction* (3rd ed.). Reading, MA: Addison-Wesley.

Skelton, T. M. (1992). Testing the usability of usability testing. *Technical Communication, 39*(3), 343–359.

Skinner, B. F. (1956). A case history in scientific method. *American Psychologist, 11*, 221–233.

Sklar, A., & Gilmore, D. (2004). Are you positive? *Interactions, 11*, 28–33.

Smith, P. A. (1996). Towards a practical measure of hypertext usability. *Interacting with Computers, 8*(4), 365–381.

Snyder, C. (2003). *Paper prototyping.* New York: Morgan Kaufmann Publishers.

Snyder, C. (2004). Paper prototyping grows up. *Proceedings of the Society for Technical Communication 51st Annual Conference* (pp. 334–338). Salt Lake City, UT.

Soderston, C. (1985). The usability edit: A new level. *Technical Communication, 32*(1), 16–18.

Swierenga, S. J., & Guy, T. (2003). Session logistics for usability testing of users with disabilities. *The Usability Professionals' Association Annual Meeting* (pp. 1–6). Scottsdale, AZ.

Tedesco, D., McNulty, M., & Tullis, T. (2005). Usability testing with older adults. *The Usability Professionals' Association Annual Meeting* (pp. 1–8). Montreal, CA.

Theofanos, M., & Quesenbery, S. (2005). Towards the design of effective formative test reports. *Journal of Usability Studies, 1*, 27–45.

Thomas, B. (1996). Quick and dirty usability tests. In P. Jordan, B. Thomas, B. Weerdmeester, & I. McClelland (Eds.), *Usability evaluation in industry* (pp. 107–114). London: Taylor & Francis.

Tullis, T., Flieschman, S., McNulty, M., Cianchette, C., & Bergel, M. (2002). An empirical comparison of lab and remote usability testing of websites. *The Usability Professionals' Association Annual Meeting* (pp. 1–5). Orlando, FL.

Tullis, T., & Stetson, J. (2004). A comparison of questionnaires for assessing website usability. *Proceedings of Usability Professionals' Association* (pp. 1–12). Minneapolis, MN.

Urokohara, H., Tanaka, K., Furuta, K., Honda, M., & Kurosu, M. (2000). NEM: "Novice Expert ratio Method" A usability evaluation method to generate a new performance measure. *Proceedings of Human Factors in Computing Systems* (*CHI 2000*) (pp. 185–186). The Hague, Netherlands.

Vatrapu, R., & Pérez-Quiñones, M. A. (2004). *Culture and international usability testing: The effects of culture in structured interviews* (Technical report cs.HC/0405045). Retrieved October 4, 2005, from Computing Research Repository (CoRR) website: perez.cs.vt.edu/publications.php

Virzi, R. A. (1989). What can you learn from a low-fidelity prototype? *Proceedings of the Human Factors Society, 33rd Annual Meeting* (pp. 224–228). Denver, CO.

Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. *Proceedings of the Human Factors Society, 34th Annual Meeting* (pp. 291–294). Santa Monica, CA.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors, 34*, 457–468.

Virzi, R. A., Sokolov, J. L., & Karis, D. (1996). Usability problem identification using both low- and high-fidelity prototypes. *Proceedings of Human Factors in Computing Systems* (*CHI'96*) (pp. 236–243). Vancouver, Canada.

Virzi, R. A., Sorce, J. F., & Herbert, L. B. (1993), A comparison of three usability evaluation methods: Heuristic, think-aloud, and performance testing, *Proceedings of the Human Factors and Ergonomics Society, 37th Annual Meeting* (pp. 309–313). Seattle, WA.

Vora, P. (1994, November). Using teaching methods for usability evaluations. *Common Ground, 4*, 5–9.

Walker, M., Takayama, L., & Landay, J. A. (2002). High-fidelity or low-fidelity, paper or computer? Choosing attributes when testing web prototypes. *Proceedings of the Human Factors Society, 46th Annual Meeting* (pp. 661–665). Atlanta, GA.

Waters, S., Carswell, M., Stephens, E., & Selwitz, A. (2001). Research ethics meets usability testing. *Ergonomics in Design*, 14–20.

Wichansky, A. (2000). Usability testing in 2000 and beyond. *Ergonomics, 43*(7), 998–1006.

Wiklund, M. E., Thurrott, C., & Dumas, J. S. (1992). Does the fidelity of software prototypes affect the perception of usability? *Proceedings of the Human Factors Society, 36th Annual Meeting* (pp. 399–403). Santa Monica, CA.

Wilson, C. E., & Coyne, K. P. (2001). Tracking usability issues: To bug or not to bug? *Interactions, 8*, 15–19.

Wolf, C. G. (1989). The role of laboratory experiments in HCI: Help, hindrance or ho-hum? *Proceedings of CHI'89* (pp. 265–268). Austin, TX.

Young, R., & Barnard, P. (1987). The use of scenarios in HCI research: Turbocharging the tortoise of cumulative science. *Proceedings of CHI'87* (pp. 291–296). Toronto, Canada.

·58·

SURVEY DESIGN AND IMPLEMENTATION IN HCI

A. *Ant Ozok*
UMBC

Introduction **1152**
Purpose of Survey Use and Survey Applications in HCI 1152
Brief History of Survey Design in HCI 1153
Paper-and-Pencil and Electronic Surveys 1153
Survey Design in HCI **1156**
Survey Types 1156
Opinion Surveys 1156
Evaluation Surveys 1157
Other Survey Types 1158
Survey Application Domains 1159
Survey Design Methodology 1159
Survey Design Techniques 1159
Content design 1159
Scales and open-ended questions 1160
Scale design 1160
Survey Evaluation Techniques 1161
Survey validity and validity determination 1161
Survey reliability and reliability measurement 1162
Other survey evaluation issues 1163
Survey Design, Redesign, and Revision 1164
Illustrative Survey Design Example 1164
Challenges in Survey Design 1164
Survey Implementation in HCI **1164**
Open vs. Controlled Implementation Environments 1164
Sample Representativeness in Survey Implementation 1166
Challenges in Survey Implementation 1166
Emerging and Future Trends in Survey Design and Implementation in HCI **1168**
References **1168**

INTRODUCTION

Survey and questionnaire design has been a primary source of data collection within the Human-Computer Interaction (HCI) context since the early days of the science (Baecker, Grudin, Buxton, & Greenberg, 1995). Within the HCI context, surveys are defined as compilations of questions that are implemented either via a computer or paper-and-pencil-based environment, that either have quantitative or qualitative scales, or are open-ended, and that target at extracting a variety of information from a representative sample of the target population (which is in most cases current or prospective users of an HCI system being evaluated).

Survey use is popular in HCI research as it allows researchers to collect, in a relatively easy manner, information based on users' satisfaction, opinions, ideas, and evaluations regarding a system. Design and implementation of surveys are not as costly as conducting experiments in closed environments with special equipment; advances in computer-based survey products and web-based survey services allow direct recording and easy manipulation of survey data by eliminating the need of translation from paper-based to electronic environment; and, with each survey taking minutes to complete in most cases, given a large sample of potential participants can be reached, surveys are a good resource for collecting large amounts of data in a relatively short amount of time and with minimal resources, especially when compared to controlled objective experimental measures that involve in most cases lengthy tasks and recording sessions. On the other hand, surveys are constantly challenged in terms of their validity and reliability mostly due to their high reliance on participant opinions and the impossibility to measure with full reliability that the questions are answered by participants objectively. Quantitative survey research is also sometimes criticized due to difficulties related to survey scaling, as scales rely on an assumption that participants have the same or similar perceptions of scale responses that are subjective in structure (responses such as "I strongly agree" or "I often do it").

This chapter discusses the different aspects of survey design and implementation in HCI in a structured and comprehensive manner. After a discussion of the purpose and a brief history of surveys, the different types of surveys (content and structure-wise), application domains, design, and evaluation techniques are discussed with illustrative examples. The chapter is concluded with emerging and future trends in the HCI survey design and implementation areas.

Purpose of Survey Use and Survey Applications in HCI

Usability evaluation has been a primary component of HCI since its inception in 1960s. User performance and satisfaction have long been tapped as the major components of usability testing and evaluation (Shneiderman, 1992; Nielsen, 1989). While user-performance measurement relies on objective methods such as software-based time and error measurement, user satisfaction requires more sophisticated tools to be measured objectively. User satisfaction is defined as the level to which a system meets its user's standards and requirements (Hackman & Oldham, 1980).

Directly relating to user satisfaction, user preferences, opinions, and evaluations concerning HCI systems are also of strong interest to usability testing and evaluation processes. Since it is not possible to measure all of these usability components through unequivocal measurement methods, explicit tools have been developed to elicit information regarding user satisfaction, preferences, opinions, and evaluations both qualitatively and quantitatively through user surveys. Surveys serve this specific purpose well by posing targeted questions to users for the purposes of HCI design and evaluation.

While surveys can be designed to collect a variety of types of information concerning the target population, relevant to HCI research and literature (Card, 1996), they are mostly targeted at collecting information in the following three categories:

a. **User Evaluation.** The category aims at collecting information regarding how much a system, product, or environment meets user goals, expectations, and standards. In this category, users are asked a number of questions regarding whether their overall impression regarding the object being evaluated is high, what exactly constitutes this impression, what and where exactly the problems are, and so on. Relating to user satisfaction, this category is also about determining user opinions specific to products or systems, where questions can also include users' opinions concerning whether tasks can be completed effectively and efficiently, whether the system is fast, and so forth.
b. **User Opinion.** The category can, but does not have to, be specific to products, systems, or environments. These types of surveys are aimed at determining what users think about the requirements from a system, product, or environment to fulfill its function satisfactorily. Examples can include surveys that aim at needs assessments for the next generation of cell phones (e.g., what new functionalities can be useful in newer cell phones besides those that already exist according to cell phone customers). Simply put, while the former category consists of surveys regarding the performance of existing systems, environments, and products, the current category is concerned with what users think about what might be useful concerning these systems in more general terms.
c. **Others.** The third category includes the remaining possible survey types aimed at collecting a number of different information types within the HCI context. One such category consists of surveys that are strictly concentrated on population demographics. These types of surveys do not contain questions relying on participants' evaluation of specific products or their opinions, but rather solely on qualifications they own, such as age, sex, education level, skill level, and so forth, or things they do, such as how frequently they go on the Internet or use a cell phone. These types of survey questions are less based on opinion-heavy responses than the previous two categories.

Brief History of Survey Design in HCI

Surveys started being used as a computer science and, to a limited extent, an HCI research tool in early 1970s, borrowing techniques from anthropology and experimental and social psychology (Myers, Hollan, & Cruz, 1996). With contribution from developments in the overall survey administration and language issues (Converse & Presser, 1986; Belson, 1981; Jenkins & Dillman, 1997), researchers discovered early on that information regarding user attitudes, preferences, and evaluations within the context of computer technology (software and hardware) development can be collected fairly easily with paper-and-pencil surveys. Hence, in the 1970s and 1980s, user surveys were a part of computer research with a social psychology emphasis, but not directly relating to usability testing and usability-design issues. In the mid-1980s graphical user interfaces became an important part of end user computing and usability research took off. The surveys gained a more significant role in HCI research around the same time, and with the advent of graphical user interfaces, surveys in HCI and specifically usability research gradually gained importance. With the graphical user interfaces as we know today gaining high popularity with Windows 95, usability research accelerated (Myers, 1996), and besides building usability laboratories, companies and research institutions started developing and implementing surveys to determine user trends and preferences in HCI. Advanced electronic and paper-and-pencil survey-design methods have been developed in the last decade (Dillman, 2000), and user surveys have become an essential part of data collection in HCI research.

Paper-and-Pencil and Electronic Surveys

Survey implementation largely relies on practical factors. Besides the challenge of finding a sample size that is both large enough and representative of the population the study is targeted at, implementation challenges include presenting the participant sample with a survey that is quick and easy to complete, has a fun factor and a learning component for the participants, and does not require tedious data extraction and manipulation once implementation is completed. Additionally, surveys in every topic should be unambiguous, unbiased, correctly coded, piloted, and ethical (Stone, 1993). Today's surveys are almost universally implemented in two forms: (a) paper-and pencil surveys, which require participants to mark or write their responses on response sheets that also contain questions, either on the same sheet or separately; and (b) electronic surveys, which require the users to use the screen, keyboard, and/or mouse of a computer to mark or type their responses on the screen.

Paper-and-pencil-based surveys require the survey material to be physically distributed, filled out, and returned. This process can occasionally be cumbersome and tedious. Moreover, these types of surveys also require manual entry of quantitative, and in most cases qualitative data, for analysis. One solution to the problem of translation of paper-based data into electronic format is using Scantron sheets which are sometimes referred to as "bubble-sheets." In this system, designated areas (bubbles) on printed sheets are filled out by participants with a pencil, and these sheets are then fed into a computer and scanned for correct answers. This process, however, is quite costly due to the scanning equipment necessary for the process. Figure 58.1 shows two sample paper-and-pencil survey sheets, one with response spaces below each question and one with response sheets that are separate from question sheets.

Although about 62% of all American households own one or more computers (U.S. Census Bureau, 2005), computers are still not accessible to the entire population. Therefore, paper-and-pencil surveys are still widely popular. Paper-and-pencil surveys allow swift collection of large data quantities if they are administered to groups of participants simultaneously, such as a group of students during a class period who are asked to fill out and return the surveys immediately. One other common way of paper-and-pencil-based survey implementation is mailing in surveys to participants and asking them to return them via mail, in most cases in postage-prepaid envelopes. However, recent studies indicated that return rates of mailed-in surveys by the participants is highly variable depending on the type of survey (Dillman, 1991, 2000). Taking into consideration the percentage rate of computer ownership among American households, mailed-in surveys can be concluded as a less-preferred means of data collection specifically within the context of the HCI research, and mail surveys are therefore not popularly used for HCI research purposes.

Computer-based surveys (sometimes referred to as PC-based surveys) have become popular with the advent of home computers in the 1990s, even before the high adoption of the World Wide Web. In computer-based surveys, participants are presented the survey on a specific, standalone (non-networked) computer. It should be noted that these types of surveys are not web-based, but rely on the software installed on the computer on which the survey is implemented.

Participants use the mouse to click on their responses of choice for multiple-choice questions. Mouse actions are generally implemented on dropdown combo boxes (where a dropdown menu opens up with the options when the user clicks on the button located to the right side of it) or radio buttons (a round area to the left of the option is clickable), or check boxes where multiple choices can be clicked on at one time (a square-shaped box to the left of the option is clickable) (Ozok & Salvendy, 2000). For text entries, participants can type text on specified text boxes. While computer-based surveys can be convenient because of having the initial data in electronic format and eliminating the necessary transformation to electronic format in paper-and-pencil-based surveys, they require the participants to be stationary on a specific computer. For large-size implementations, computer-based surveys can be extremely slow in collecting the data, mainly due to limited computer equipment and scheduling difficulties. It can be concluded that while computer-based surveys can be advantageous in the data-analysis stage and are still popular in kiosks stationed in public places such as shopping malls, they are not suitable for large-sample size and lengthy surveys, and hence are not the best solution in survey-based data collection in HCI context.

With the advent of the Internet, web-based (online) surveys have become highly popular (Dillman, 1999) and allowed researchers to conduct surveys more effectively and efficiently than

1. My cell phone screen is convenient to use.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

2. The keypad on my cell phone is a convenient means of data input.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

3. My cell phone interface is easy to use.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

Question Sheet:

1. My cell phone screen is convenient to use.
2. The keypad on my cell phone is a convenient means of data input.
3. My cell phone interface is easy to use.

Response Sheet:

	○	○	○	○	○	○	○
1.	Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree
	○	○	○	○	○	○	○
2.	Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree
	○	○	○	○	○	○	○
3	Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

FIGURE 58.1. Two paper-and-pencil presentations of the same survey, one with the response spaces below each question and one with separate question and response sheets.

more traditional means (Zhang, 2000). Server-based software allows survey participants to access a web page and fill out the survey, then submit their results mostly to a central server by clicking an on-screen button. web-based survey interfaces in structure look very similar to computer-based surveys with same interface elements of dropdown combo boxes, radio buttons, check boxes, and text boxes serving the functions of various data-entry types by the participants. The data are collected on a central web server in these types of surveys, which can be easily obtained and manipulated by the survey administrators. Additional data storage and analysis programs residing on these web servers can compile the data in a variety of formats such as Microsoft Excel, and also implement some automatic data analyses such as calculation of means and standard deviations (descriptive statistics).

One additional electronic survey type consists of the administration of e-mail surveys in which participants are e-mailed a survey and are asked to fill it out and e-mail it back. However, with the dramatic increase in the amount of spam e-mail users receive in recent years, it can be concluded that these kinds of e-mails are likely to be perceived as spam and are likely to be discarded. Therefore, e-mail surveys are not articulately covered in this chapter.

There are two types of methods used in web-based survey administration today. One method is to use a web survey administration service provider (such as SurveyMonkey.com) by paying it a monthly or yearly fee. In most cases, various packages are available ranging from a small number of surveys to unlimited administration of surveys. The web service providers also have a variety of options for the retrieval of survey data by the administrators, for example in Access or Excel formats, with some, in most cases basic, statistical analyses (such as descriptive statistics and correlations) already performed on the data. Additionally, the services also give flexibility in customization of survey interfaces such as giving the freedom to the administra-

tors (and in some rare cases to the participants) to choose their text and background colors, font sizes and types, how many questions to have per web page, and so on. Today, there are more than a hundred online survey-service providers with monthly fees varying from $3 for small-scale academic surveys to $500 for large-scale, mostly marketing-related surveys.

Another means of administering online surveys is to use one's own hardware and software. As a central server is necessary for collection and storage of the data, this hardware can either be purchased or rented, or an existing server can be used. The amount of storage space necessary largely depends on the length and sample size of the survey, but since most surveys contain textual data, it is almost unimaginable to need more than five gigabytes of storage space for a large-size survey for purposes of HCI research. A variety of open source (such as php ESP [Easy Survey Package]) and licensed software vendors (such as Inquisite) are available for survey implementation and administration on administrator-owned servers. Like survey-service providers, these software packages also allow a variety of customization flexibilities concerning the survey interfaces. Figure 58.2 presents a sample interface from an online survey.

For both paper-and-pencil-based and electronic surveys, human cognitive limitations should be taken into consideration and basic human factors guidelines should apply. Paper-and-pencil surveys should be administered with pencil to allow participants to go back and change their responses if they want to. White paper with black, 12–14-point-sized Times New Roman font text should be used, as those font sizes are the most common and easily readable text sizes in printed documents. Labels should be presented with bold, 16–18-point-size text, and while pictures are seldom presented to participants, if they are presented, they should have enough resolution to appear clearly on paper. In short, paper-and-pencil-based survey interfaces should be inspected to make sure they comply with the structural human factors guidelines for paper-based interfaces.

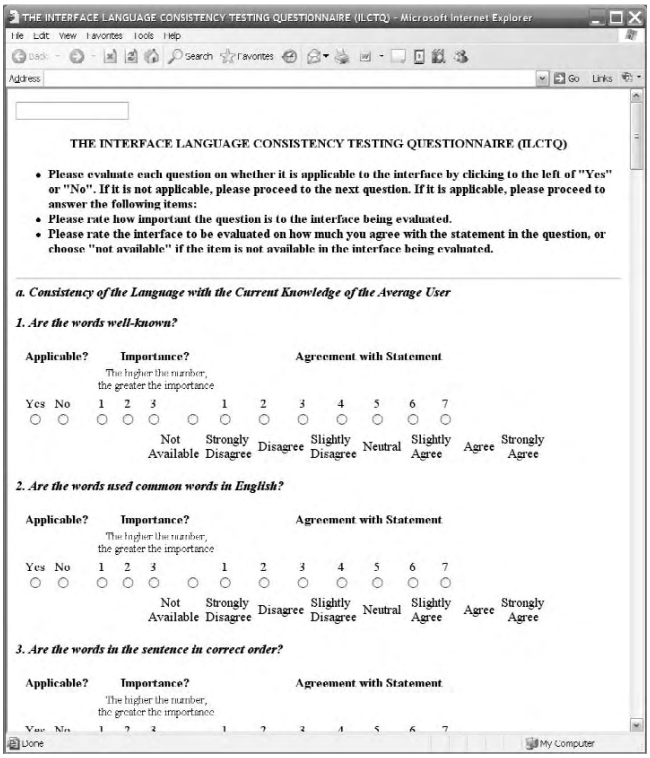


FIGURE 58.2. Sample interface from an online survey.

Similarly, electronically administered surveys (web- or PC-based) also should follow basic human-factors guidelines for computer interfaces. For web-based surveys, it should be noted that participants will access the surveys from a variety of different types of machines, and basic computer and web design guidelines indicate 800×600 color screen resolution is the most common screen type (Nielsen, 1993; Badre, 2002) which should be taken into consideration when survey interfaces are designed, making sure that survey objects, mainly text fonts and some images if there are any, are easily visible on screens with this resolution. The basic rule of black text on white background in web usability should also be applied in computer- and web-based interfaces, and screen objects such as dropdown combo boxes, radio buttons, buttons, and check boxes should be the same size (mainly height) as text lines to ensure consistency and easy visibility and to allow users to click on them easily. Text boxes for the users to type in should allow text size consistent with the text on the question parts of the interface. Overall, again, it can be said that in web-based surveys, basic web-design guidelines can easily be adopted.

Additionally, one important item in survey design is to prevent survey participants from getting worried about the excessive length of the surveys, as too-long surveys may result in significantly decreased response rates (Lund & Gram, 1998; Krosnick, 1999). Taking also into consideration the fact that computer users don't enjoy scrolling down the screen in general (Shneiderman, 1992), no more than 20 questions should be presented on one survey screen. For surveys containing more than 20 questions, subsequent questions should be presented on subsequent screens (again, each screen not exceeding 20 questions), which participants should move to by clicking on a screen link or button that should have a statement such as "Click here to continue."

Both paper-and-pencil-based and electronic surveys will continue to be used in HCI research. While it is unlikely that the former will go extinct any time soon, recent studies (such as Dillman, 2000) showed that web-based surveys have very significant advantages in data collection and analysis stages of survey-based research. The fact that most HCI-related survey research uses computer-savvy sample participant groups is also a factor that helps the popularity of web-based surveys within HCI. It is therefore expected that web surveys will eventually constitute to a large majority of survey-based research in HCI. While no previous study explicitly investigated the exact share percentage of paper-and-pencil-based and electronic surveys in HCI research, it is estimated that more than 60% of all surveys for HCI research are implemented on the web environment.

While the discussion in this chapter mostly concentrated on structural issues of paper-and-pencil-based and electronic survey design, a much larger research topic concentrates on how the actual survey content should be designed, created,

and implemented. The next chapter discusses content creation in surveys within the context of HCI research.

SURVEY DESIGN IN HCI

A major part of survey-design research in HCI is concerned with the formulation of survey questions and scales. In comprehensive user surveys, questions may be presented as one large, continuous list. Related questions may also be presented in categories and sections, such as the demographic questions in the first section, questions evaluating the structural elements of the interface in the second section, and so on. As discussed earlier, surveys in HCI research are mostly concentrated on collecting two categories of information: participants' opinions and evaluations. These two survey types are discussed in the next section. This discussion is followed by an explanation of survey application domains and the survey design methodology.

Survey Types

Opinion Surveys

Opinion surveys aim at determining what participants think about issues pertaining to certain concepts. Additionally, opinion surveys can also measure participants' wishes, habits, and customs (Baker, 1998). The way opinion surveys differ from evaluation surveys is that opinion surveys are not centered on a specific product or environment, but are more general in nature. For example, an opinion survey can target measuring cell phone users' habits and opinions regarding the use of the cell phone interface (whether they are happy with the screen size, whether they can easily enter data using the keypad, etc.). While questions are focused on a product in this example (e.g., a cell phone) the population of cell phone users is highly diverse, as are cell phone devices that are used by the population. Hence, such an opinion survey will measure general trends in usability concerning cell-phone interfaces rather than evaluating a specific cell phone interface.

Opinion surveys are administered with more freedom than evaluation surveys, as the participants do not have to have the evaluated issue, product, or environment fresh in their memory. Hence, they can fill out opinion surveys at any time and in any environment that is convenient for them. Opinion surveys include questions that do not require any recalling process, meaning they contain questions which participants can answer without having to recall a specific feature or part of the environment, product, or issue. As opinion surveys ask general questions about participants' current standing regarding opinions, wishes, customs, and habits, the answers may sound subjective and vary greatly among participants. Sample questions for opinion surveys include statements such as "Does the use of computers in daily tasks contribute to your overall technology knowledge?" or "Are you in general satisfied with the amount of product information available on an e-commerce page you frequently visit?" or "How would you rate the customer services on the sites where you frequently shop?"

Opinion surveys can cover a broader variety of issues than evaluation surveys which are more focused. They can include both qualitative and quantitative scales for their responses. Although no previous literature came up with a strict classification of opinion surveys, the following classification can help in structuring of opinion surveys and what kind of questions should be asked for what types of opinion-related survey categories in HCI research (Sample questions for each classification of opinions are presented on Table 58.1):

a. **Opinions on a medium.** Within the context of HCI research, these types of surveys concentrate in most cases on interface design, usability, user satisfaction, and user preferences issues concerning a medium the participants use in their daily lives. Most popular examples of this type of media include daily-used devices such as computers, Personal Digital Assistants (PDAs), and cell phones. These types of surveys concerning users' opinions on a certain medium aim at determining general trends in user opinions concerning whether the design of the medium is satisfactory and meets user needs and requirements, whether there are problems concerning the medium, and what can be possible solutions. Opinion surveys concerning a medium are also useful when they are used by usability specialists and engineers to develop new versions of products or to come up with new products, as the survey results can pinpoint the needs of the target population to be met regarding the medium. Sample questions for these types of surveys can include questions like "Are you in general satisfied with the web browsing capabilities of your cell phone?" or "What additional capabilities would you like to see on your PDA user interface?"
b. **Opinions on an event.** Within the context of HCI research, user opinions on an HCI-related event can include what they think about certain HCI-related happenings. Examples can include opinions concerning certain HCI-related activities, with questions like "Do you find the move from a text-based interface to a graphical user interface helpful?" These types of surveys are rarer in nature and aim at collecting basic trends concerning user opinions in current HCI-related activities.
c. **Opinions on a procedure.** These kinds of surveys aim at determining the user opinions on procedures to complete HCI-related tasks. They are similar to medium-related opinion surveys, but rather than questions about the medium itself, these surveys have the goal of determining user opinions on how it is used. In web and e-commerce design, these kinds of opinion surveys are helpful in determining whether the procedures to complete general tasks (for example, web navigation) meet user requirements and needs. A sample question in an opinion survey concerning a procedure could be "Are you satisfied with how long it generally takes to purchase a product on an e-commerce site?" Surveys to explore opinions on a procedure are less common as HCI researchers usually resort to evaluation surveys to test procedures for use of computer interfaces and other media (explained in the next section).

TABLE 58.1. Opinion and Evaluation Survey Sample Questions

Opinions on a Medium:

1. Personal Digital Assistants (PDAs) are convenient.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

2. I would prefer using a laptop to a desktop computer for my daily tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

3. Tablet PCs are capable to meet my daily computing needs.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

4. Setting up a wireless Internet connection is easy on a computer.

○	○	○	○	O	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

Opinions on an Event:

1. Free wireless area networks should be universal.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

2. Advancements in Liquid Crystal Display (LCD) technology made them more affordable.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

3. Free services on the Internet meet my expectations.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

4. The Windows computer interface is significantly easier to use now than it was five years ago.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

Opinions on a Procedure:

1. The sign-up process for e-commerce sites is too time-consuming.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

2. Password authentication procedures are reliable.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

3. The steps involved in personalizing a cell phone interface settings are long.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

4. Payment transaction procedures on the Internet are mostly secure.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

Evaluation (of a Computer Interface):

1. How difficult did you find completing the tasks on the interfaces?

○	○	○	○	○	○	○
Extremely Difficult	Difficult	Somehow Difficult	Neither Easy Nor Difficult	Somehow Easy	Easy	Extremely Easy

2. How intuitive were the interfaces you used?

○	○	○	○	○
Not Intuitive at All	Not Intuitive	Neutral	Intuitive	Very Intuitive

3. While completing the tasks, I made _____.

○	○	○	○	○
No Errors	Few Errors	Neither Many nor Few Errors	Quite a Few Errors	Many Errors

4. Were the interfaces aesthetically pleasing?

○	○	○	○	○
Not at All	Very Little	Neutral	Quite a Bit	A Lot

Evaluation Surveys

More specific than opinion surveys, evaluation surveys (or questionnaires) are generally administered after a certain procedure is implemented on the participant group. While opinion surveys can be administered at any time to the participants, evaluation surveys are administered right after the participants have completed a certain procedure. In evaluation surveys, participants are asked about tasks they have just recently completed. Therefore, evaluation surveys are in most cases preceded by certain usability-related experimental procedures. Most common in HCI, evaluation surveys are administered after participants have completed a number of tasks in controlled computer environments. They are also implemented right after the procedure to ensure that memories regarding the procedure are still fresh in the participants' minds, as evaluation

surveys require a high amount of recall of procedures and interfaces in the tasks that were just previously completed.

HCI-related evaluation surveys have the main goal of evaluating usability, user satisfaction and user preference issues concerning user interfaces or environments (Ozok & Salvendy, 2001). After certain tasks are completed in these controlled environments, evaluation surveys aim at determining the exact nature and location of problems and points open to improvement in the human-computer environments. Therefore, evaluation surveys are in most cases relatively detailed in nature. In a sample procedure for example, participants can be presented a number of web pages and asked to complete some common user tasks on those pages, tasks that can include text entry, direct manipulation, and form filling. An evaluation survey that would follow could include questions such as "Was the site navigation difficult?", "Was the text size on the site readable?", "Did you enjoy the overall experience on this site?", "Would you visit this site again?", and so on.

As their name indicates, evaluation surveys aim at evaluating interfaces, environments, and procedures from the user's perspective. For that purpose, they are tools to determine how participants evaluated those interfaces, environments, and procedures. In that sense, evaluation surveys are explicit and not that much different from objective measurement methods such as performance measurement in HCI tasks. Similar to opinion surveys, evaluation surveys use both qualitative and quantitative scales.

Evaluation surveys are not only helpful in evaluation of product interfaces, environments, and procedures. They can also be used in evaluating certain conceptual features. For example, a researcher may be investigating whether the introduction of a certain interface concept results in higher user satisfaction regarding the interface. If this particular feature is, say, that of interface visibility, the survey administrator can first present the participants with high-visibility and low-visibility computer interfaces, and the subsequent evaluation survey can contain questions such as: "Did higher visibility improve your overall satisfaction with the computer screens?"

Evaluation surveys are useful in a variety of HCI-related commercial and academic research activities. In academic research, different computer or computer-related interfaces can be evaluated through surveys to determine whether they result in better user preferences or higher user satisfaction. In commercial research, newly developed products (for example a new computer peripheral), environments (for example a new computer interface), or procedures (for example the steps it takes to complete a transaction with a new e-commerce site design) can be empirically evaluated by having participants complete tasks with those products and procedures or in those environments, then filling out evaluation surveys consisting of detailed questions regarding their satisfaction and preferences regarding the said product, environment, or procedure. While quantitative evaluation results can give statistical backing to user evaluations, helping boost their conclusiveness, qualitative evaluation results can give the researchers who administered the surveys new ideas to improve usability/user-preferences-related design components. Therefore, evaluation surveys are the most commonly used survey types in HCI research and constitute to one of the most common and effective user evaluation methods in HCI in general. Examples of opinion survey questions are also presented in Table 58.1.

Other Survey Types

Besides the two main survey types mentioned earlier, one widely used survey type is the demographic survey. Although demographic surveys are almost universal, it would be incorrect to categorize them at the same level as opinion and evaluation surveys, as most user surveys have a section concerning user demographics. Hence, in most cases demographic survey are essential parts of opinion and evaluation surveys rather than stand-alone surveys. In some cases HCI researchers administer surveys consisting of demographic questions only—for example, to determine the demographics of a user group using a specific application. However, more commonly, HCI surveys consist of opinion and/or evaluation questions in addition to the demographic questions.

Demographic questions play an important part in HCI-related survey design, as most variables of interest in HCI research are also dependent on the factors that are specific to the target population. Consequently, research findings can only be generalized in most cases to the target population from which a representative survey is sampled.

Demographic surveys (or survey sections) in most cases consist of a standard set of categories: age, sex, education, and occupation of the participant. Age can also be asked in the form of "birth year," and for the question regarding sex, options in the form of "Male/Female" can be presented to participants to mark on the computer or with the pen/pencil. Education level can be left to type in or write, or options can be presented. The question is usually formulated as: "What is the highest degree for which you won a diploma?" or "What is the highest degree you earned?" Typical options for this question are "Elementary School," "Middle School," "High School," "College or University," "Graduate Degree," and "Post-Doctoral Graduate Degree." The occupation question is about what kind of job the participant has. For this question, usually the participants are asked to type or write in the designated area, due to the high variety of possible occupations participants may have, although presenting options for this question is also possible if some general occupation categories are all that is needed, for example options such as "Private sector," "Academia," "Student," and so on.

In addition to this basic set of survey questions, demographic surveys can also include general questions regarding daily habits or current standing issues concerning the participants. Most commonly, demographic surveys in the HCI area contain questions regarding computer use habits of participants, such as "How many times a day do you check your e-mail?" or "How many times in the last year did you shop from a web-based e-commerce company?" These types of questions are usually customized according to the type of information needed for the specific research being conducted and can therefore greatly vary in nature, but in principle they aim at collecting information on computing-related habits in most cases. Table 58.2 presents a set of sample demographic questions as part of a survey regarding cell phone use.

TABLE 58.2. Sample Demographic Questions from a Survey on Cell Phone Use

Your Age:
Your Gender:
Your Occupation:
How Many times a week do you go on the web?:
_____ Less than once a week
_____ Between once and three times
_____ Between three times a week and every day
_____ Every day
In the past year, how many times did you shop online (Please put a number)?: ____
Do you own a cell phone, a Personal Digital Assistant, or a Combination Device?
_____ Yes
_____ No
In the past one year, how many times did you shop online using a cell phone, a Personal Digital Assistant or a Combination Device (Please put a number)?: _____

It should be noted that demographic questions are of a more personal nature than opinion and evaluation questions. Some participants may feel that their privacy rights are being violated by being asked to provide their age and education level. For this reason, providing anonymity in surveys and informing the participants about their provided information being not personally identifiable—in other words, providing anonymity—is greatly crucial in HCI research. Knowing that their data cannot personally identify them usually takes care of privacy worries and is known to improve participant enthusiasm. Challenges concerning privacy in survey implementation are described later in this chapter.

There are no other significant survey types widely used in HCI research. Some niche survey types may still exist but are few and far between. Therefore, surveys aimed at collecting information relevant to HCI research usually belong to one of the categories explained in this section. In the next section, application domains of surveys relating to HCI research are discussed.

Survey Application Domains

Survey applications are highly popular in a broad range of application domains, in areas ranging from social sciences to marketing, education to production, customer to worker satisfaction, and many more. Today, results obtained from surveys which in most cases ask comprehensive questions are deemed reliable and valid in both scientific and industrial projects. The most common application domains of survey research include

a. **Sales and Marketing:** Companies that offer products and services for both consumers and industries use customer/client surveys for both needs-assessment and evaluation purposes. A large number of companies are also solely dedicated to implement customer surveys for companies, analyze the data, and deduct conclusions for sales and marketing purposes. Customer satisfaction, product evaluation, customer relationship management, and customer demographics are only a few of the topics surveyed by sales and marketing forces.
b. **Medicine:** Medical research is not limited to trials relying on objective measurements. Surveys can be helpful in collecting patient data for development of medicine or treatments.
c. **Education:** Educational surveys can help determine population preferences in education as well as education levels and education-related difficulties among population segments.
d. **Information Technology Research:** In the field of information technology, surveys are widely used in connection to software and hardware design and evaluation, covering a broad variety of areas including software engineering, systems analysis and design, and of course human-computer interaction, which this chapter covers.

The earlier-mentioned relevant list covering the application domains of surveys is far from complete, but a sample of application domains are presented in the list. The list will no doubt continue growing with the advent of new technologies and sciences. The HCI area is seen as a major application domain of surveys, and is expected to continue to be so.

Survey Design Methodology

Survey design is a methodological activity that requires a systematic design procedure. Survey design techniques and procedures are discussed in this section, including content design and scale design, followed by survey design and redesign issues, a survey design example, and a discussion of challenges in survey design.

Survey Design Techniques

Survey design mainly consists of two components: the design of the survey content and the survey scale. They are both discussed in this section.

Content design. In the heart of the survey research lays the issue of producing the actual questions to ask the participants. Designing the survey content is actually producing these questions along with their scales. Deciding on which questions to ask to the participants in the survey largely depends on three resources: literature, expert opinions, and individual experiences.

A large number of survey questions are based on previous research in the focus area. Relying to some extent on previous literature allows the researchers to achieve high validity of their survey structure, as previously validated research allows current survey design to have strong backing in terms of its content and the targeting of the questions concerning the particular

research topic. Therefore, it is best to have backing from previous studies at least for the majority of the questions while designing the survey.

While a designed survey's content may consist largely of questions that are based on the relevant literature in the area, there will be most likely some issues that are intended to be included in the survey but are not covered in the previous literature. Therefore, HCI researchers sometimes rely on experts in the area to cover additional points to be included in the survey. A preliminary survey may be sent in this context to the area specialists to determine the most significant items to be covered in the investigated area. For example, if a survey research is trying to determine the most significant interface design items in e-commerce that affect buying behavior, a preliminary survey may be sent to experts in the area (for example, e-commerce company managers and professors in business schools specializing in e-commerce) to determine the general classifications of interface issues relating to the buying decision. In addition to the literature, these responses from experts can be used as a major resource of question generation for the resulting survey.

Researchers can also rely on their own heuristics and expertise in producing questions. To prevent being accused of "making up the questions," the researchers would need to explain logically why the questions were included in the survey. In these cases, researchers can include questions based on what they think is a significant part of the research item being investigated, or based on the impression that, although the literature did not explicitly point out the issues in these types of questions, there was an implicit indication in previous research towards this particular direction.

Design of survey content is not a difficult task once the researcher has a reasonable background in the area of interest. One common mistake done in design of survey content is the researchers missing important questions during the design and ultimately not addressing those questions. Therefore, cautious, repeated reviews and revisions are necessary before the final implementation of the survey.

Scales and open-ended questions. Just as important as the content of the questions, the scales for the survey questions in HCI are essential for the accuracy and validity of survey results. Scales are created to attribute numerical values to participant responses in the survey, thereby allowing statistical analyses and giving statistical backing to conclusions obtained from the research. To respond to a scaled question, the participant marks one of the several options, the option which best represents his or her opinion regarding the item in the question. If the question has a large variety of possible answers, or if it requires a lengthy answer, then an open-ended response style may be preferred rather than presenting a scale to the participant. For open-ended responses, participants are mostly given the freedom to write or type as much as they would like.

Both scaled and open-ended questions are suitable for different question types and types of information being obtained from the participants in the survey. Quantitative studies have to use numerical scales to statistically test their hypotheses and support their findings. Qualitative research, on the other hand, analyzes survey data without the involvement of numbers. Because participants have a much higher degree of freedom when responding to open-ended questions, qualitative responses are not restricted to the defined response universe determined by survey designers (also referred to as "survey authors"). On the other hand, conclusions derived from qualitative responses may be more arguable because they cannot be tested statistically.

One more type of responses in HCI surveys includes participants being given the freedom to mark more than one response choice. While these types of responses are generally not assigned numerical scales, these types of responses are presented in demographic surveys. In these types of questions with possible multiple responses and in open-ended questions, it is useful to present the option of "Other (please specify):" to the participants, as there is always a possibility that the survey designers may not present the option which the participant would like to give as a response. A sample question of this sort could be "Where do you generally access the Internet?" with the possible responses "Home," "Work," "School," "Coffee Shop," "Internet Café," and "Other (please specify)."

It should be noted that one alternative to open-ended survey questions are interviews and focus groups, and these more interactive data-gathering techniques are likely to result in the collection of richer data than open-ended survey questions as they allow real-time interactions between the researchers and participants. Therefore, it is not highly common in HCI research to use surveys with open-ended questions only. In most cases, especially in quantitative survey research, a mix of both open-ended and scaled questions often proves to provide best empirical results. Due to their higher frequency of use, this book chapter is more focused on the design and implementation of scaled surveys rather than surveys with open-ended questions. As part of this direction, scale design is discussed in the next section.

Scale design. While a large variety of scaling techniques are available for surveys in sociology and psychology research, HCI surveys mostly rely on Likert scales (Medsker & Campion, 1997). While contrast scales consisting of yes-or-no questions with 1/0 corresponding scales are also used, five and seven-point Likert scales are highly common (Aiken, Lewis, & Lewis, 1996). In most cases, a scale needs to consist of and odd-number of options. This way, the option in the middle can correspond to "no preference" or neutral" opinion (Dillman, 2000). Each response on the scale attributed a number to allow the researchers to conduct statistical analysis on the collected data. Item scales need to be kept consistent during the data analysis phase, meaning items should be lined up in the same direction, whether they are positive or negative—in most cases positive responses scoring high and negative responses scoring low on the scales. Inverted questions (questions that ask items in the opposite direction, as discussed later in this chapter) should have their scales reversed in order to keep consistency and allow correct data analysis. A large amount of data analysis mistakes in surveys usually happen because of scaling problems. For example, if the researchers forget to invert scales of reverse questions, then correlations and differences between responses will not come out correctly, resulting in lack of validity of research conclusions.

Scales can indicate a number of different issues. Some scales are concerned with user opinions while others are concerned about frequencies. Most common scale types include agree-

ment measurement ranging between "Strongly agree" and "Strongly disagree"; frequency measurement ranging between "Not at all" and "Very often"; quality opinions ranging between "Very good" and "Very poor"; opinions regarding probability ranging between "Very unlikely" and "Very likely"; and so on. It should be noted that survey scales offer a certain amount of freedom to survey designers on how to name the possible response options for their questions, and therefore scales come in many different varieties, from those measuring amounts (a lot, quite a bit, etc.) to frequencies (very often, often, etc.) to yes-or-no scales. Normally, in most cases a "Not Applicable" or "Not Available" option needs to be added to the scale. When this option is marked, this particular question of this particular subject is eliminated from the analysis. Note that this is different from giving a score of zero to that question in the analysis. Table 58.3 present a sample of possible response scale sets along with possible number correspondences to the responses.

Survey Evaluation Techniques

After the initial design of the survey questions, scales, and instructions to the survey participants on how to fill the surveys out, surveys need to be evaluated to determine whether they are measuring what the designers intended them to measure, whether they are reliable, and whether they produce valid results. Pilot testing is one common method to preliminarily address these issues. However, full evaluation of a survey can mostly happen only after substantial data have been collected with the survey as a tool. Therefore, the evaluation of a survey is conducted based on the data collected by it, and the two components of survey evaluation are the measurement of the survey's validity and reliability.

TABLE 58.3. Possible Survey Responses and Their Numerical Equivalences

Strongly Disagree	1	***Never***	1
Disagree	2	***Very Seldom***	2
Moderately Disagree	3	***Seldom***	3
Neutral	4	***Neither Seldom Nor Often***	4
Moderately Agree	5	***Somewhat Often***	5
Agree	6	***Often***	6
Strongly Agree	7	***Very Often***	7
Excellent	5	*None*	0
Good	4	*Very Few*	1
Fair	3	*Few*	2
Poor	2	*A Fair Amount*	3
Very Poor	1	*Quite a Bit*	4
Not Applicable	—	*A Lot*	5
Not Convenient at All	1	***Very Difficult***	1
Highly Inconvenient	2	***Difficult***	2
Inconvenient	3	***Not Difficult***	3
Neutral	4	***Easy***	4
Somewhat Convenient	5	***Very Easy***	5
Convenient	6	*Yes*	1
Highly convenient	7	*No*	0

Survey validity and validity determination. While the reliability of a survey is determined after the survey data has been collected, the validity of the survey has to be determined prior to the implementation. As the name implies, validity of a survey is the degree with which the survey instrument is valid in the results it produces, in other words whether the survey is measuring what it says it is measuring (Litwin, 1995). Generally, within the context of HCI, validity is covered twofold: *construct validity* indicates the degree of how much the survey is backed by previous research in its field, how solid its construct is. In general, as in every research, development of a survey needs to rely on previous research to give the tool literature backing, proving that the survey didn't come out of the imagination of the designer, but rather relies on a number of different research studies conducted by a number of different researchers. To prove the construct validity of their survey, designers need to prove the case that the questions they put into their survey are based on previous literature. Hence, in survey design, it is imperative to ensure that a majority of the questions have been implicated in the previous literature as items relevant to the current topic of interest. Without being able to prove this validity, it is not possible to make a convincing case regarding whether the survey is doing an undisputed contribution to the overall research topic of interest. However, it should be noted that it is almost impossible to provide a survey in which every single item has a full set of articles or books backing it. In most cases, some survey questions may have some indirect mention in the previous literature, and some survey questions may be solely based on the individual experience and/or opinion of the survey designers. This type of question generation is also acceptable, as long as the designers can prove that those questions are also based on solid research. In short, construct validity of a research survey in HCI aims at proving the conclusion unarguably that results obtained from this survey are on target and valid. Validity is therefore crucial to the success of the research conducted, of which the survey is a part.

Predictive validity is, simply put, the ability and power of the survey to predict correct results in repetitive use. A survey with predictive validity indicates that the results obtained from it in the current and future use have the power of predicting accurate results. For example, if a comprehensive survey has been produced measuring the usability of a website, researchers will need to prove as part of their study that once this developed instrument is administered, the results that are produced accurately reflect the usability level and attributes of a website. Additionally, the survey also needs to accurately reflect usability levels and attributes when it is administered on other participant groups for evaluation of other sites. If these capabilities of the survey can be proven by the researchers, then the survey can be said to have predictive validity.

Like construct validity, predictive validity does not have any quantitative measurement method to be used. Hence, a survey's predictive validity again relies on qualitatively proving that the survey results are based on solid research notions, and hence the results are accurate. Survey designers need to explain that the results produced from their surveys have been proven to accurately reflect current situations concerning the target population's specifications, evaluations, and opinions, and will continue to do so in future applications when it is administered

again. For this purpose, the elements of the survey need to be proven as elements that accurately predict results concerning the topic in focus. To make accurate predictions, surveys need to consist of elements that make accurate predictions themselves when their results are analyzed, and to ensure that these elements have predictive power, they need to rely on accurate literature findings and real-life situations. In short, similar to construct validity, predictive validity of a survey can be accomplished by ensuring that the survey relies on solid previous literature and the researchers' findings. Sometimes, to ensure that survey findings and questions have predictive power, they are evaluated by experts in the area prior to the implementation. Pilot testing is addressed later in this chapter.

Survey reliability and reliability measurement. Reliability of a survey is the measure of whether the survey is measuring things consistently, and whether the results obtained from the survey can be relied upon. A survey's reliability affects its validity, as a survey that is not reliable in its measurements cannot produce fully valid results.

While there are a number of quantitative reliability measurement techniques for survey design, especially in psychology and sociology, the two most common reliability measurement techniques used in HCI research are *internal* and *inter-rater* reliability techniques.

The internal reliability technique is concerned with whether the survey questions are understood by the participants the way they are intended to be understood when they were prepared by the survey designers. An internally reliable survey contains questions that are all comprehended the same way by all participants at all times in repeated measures when it is administered. A lack of internal reliability is a common phenomenon, as different participants can understand survey questions differently if they are not asked in a highly clear and unambiguous fashion. Therefore, to improve the internal reliability of surveys, designers need to make sure to use statements that are entirely clear and leave no room for interpretation on what is meant in the questions. An example of a low internal reliability survey question would be "Did you have tremendous difficulty completing the tasks on the web page?" In this question, participants who had little difficulty, no difficulty, and a moderate amount of difficulty may respond to the questions in a very similar way, resulting in confusion regarding whether the tasks were difficult or not. Additionally, if participants had difficulties in some tasks and no difficulties in the others, a question such as this may confuse the participants on what types of tasks (difficult or not difficult) they should base their response on. Obviously, in survey design it is important to be careful not to confuse the participants while they are filling out the surveys. Potential confusions can mostly occur on the participants' parts regarding what is meant by the survey question, and what the survey question is about (Cronbach, 1990). Surveys may have high construct validity, meaning they may have been designed based on solid research, but if they confuse the participants with their questions, they will obviously lack internal reliability, and consequently, predictive power.

The most commonly used measure for internal reliability of surveys is called "Cronbach's Alpha Internal Reliability Coefficient" (Cronbach, 1990). The coefficient relies on checking whether participants respond to the same question the same way when it is asked the second time, in a similar form. These types of questions are called "duplicate questions." The Cronbach's Alpha Coefficient is a correlation coefficient that determines the correlation between the duplicate questions, thereby giving an indication of whether the participants have the same understanding of a question when it is asked in a slightly different way, more than once. In many cases, the opposite, inverted form of the same question can be asked later in the survey. An example of two duplicate questions would be one question early on in the survey such as "Did you find the web design effective to complete the tasks?" and later, toward the end of the survey, "Was the web design effective to complete the tasks?" Alternatively, a question asking the same issue of web effectiveness in a reversed manner can also be posed later in the survey in an inverted question such as "Did you find the web design ineffective to complete the tasks?"

In general, one or two duplicate question pairs are put into surveys of moderate size, up to 40 questions. It may be more helpful to insert more than one pair of duplicate questions into surveys that contain more than 40 questions. Also, if the survey has sections (for example, in a survey measuring web usability, sections may include usability of colors, layout, navigation, etc.) it is recommended to have one duplicate pair of questions for each of the sections to have freedom about determining the individual internal reliabilities of each section.

The Cronbach's Alpha coefficient is a correlation coefficient that produces a value between zero and one. The correlation between the duplicate questions is measured, and if the coefficient is equal to or greater than 0.7, then a survey is accepted as having high internal reliability (Cronbach, 1990). A set of duplicate questions, another set of duplicate, inverted questions, and a sample Cronbach's Alpha Coefficient computer output from the SAS (Statistical Analysis Software) computer package are presented in Table 58.4.

Internal reliability cannot be measured for surveys that contain open-ended questions. In empirical HCI research, however, most surveys with quantitative parts are required to have a satisfactory internal reliability coefficient in order to prove the reliability and validity of their results. Simply put, lack of internal reliability may result in the questions not measuring what they are intending to measure. Therefore, it is imperative in HCI research to insert at least one pair of duplicate questions into quantitative surveys.

More controversial than the internal reliability measure, the inter-rater reliability of a quantitative survey is concerned about the consistency among responses given by different participants to the same question. One argument is that in objective surveys, a consistency should be expected to some level among participant responses given to the same question in the same survey. While this argument may stay true to some extent in evaluation surveys, opinion surveys, as the name indicates, are about participants' opinions, which will obviously differ from person to person. Hence, it is arguable that the inter-rater reliability coefficient is a valid measure in opinion surveys. Additionally, a certain amount of variability is always present among responses to evaluation survey questions, even if the participants are all exposed to the exact same environment prior to the implementation of the surveys. The inter-rater reliability coefficient is a cor-

TABLE 58.4. Sample Duplicate Questions and Cronbach's Alpha Internal Reliability Coefficient SAS Computer Package Output

Duplicate Pair with Same Question:

1. How would you rate the convenience of this interface?

○	○	○	○	○
Very Inconvenient	Inconvenient	Neutral	Convenient	Very Convenient

2. In general, how convenient was this interface for you to use?

○	○	○	○	○
Very Inconvenient	Inconvenient	Neutral	Convenient	Very Convenient

Duplicate Pair with Inverted Question (Inversion of the Scale Needed for Second Question):

1. How easy were the tasks?

○	○	○	○	○
Very Difficult	Difficult	Not Difficult	Easy	Very Easy

2. How difficult were the tasks?

○	○	○	○	○
Very Easy	Easy	Not Difficult	Difficult	Very Difficult

Sample SAS Output:

Variables	Alpha
Raw	0.881884
Standardized	0.882042

Cronbach Coefficient Alpha with Deleted Variable

	Raw Variables		Standardized Variables	
Deleted	Correlation		Correlation	
Variable	with Total	Alpha	with Total	Alpha
Question 1	0.717499	.	0.717499	.
Question 2	0.717499	.	0.717499	.

Pearson Correlation Coefficients, N = 272

Prob > |r| under H0: Rho = 0

	Question 1	Question 2
Question 1	1.00000	0.71750
		<.0001
Question 2	0.71750	1.00000
	<.0001	

relation coefficient among these responses given to the same question by different participants. And expecting a correlation as high as 0.7 among the participant responses may in most cases be not very realistic as a proof of reliability of a survey. Hence, while the inter-rater reliability coefficient is used in a number of survey types primarily in psychology, it is not seen as an essential measurement coefficient for survey reliability in relation to HCI research (Ozok & Salvendy, 2000). The inter-rater reliability coefficient is also used to determined how professionals in psychology and sociology rate the same situation, but this type is not covered in detail due to its lack of relevance to HCI.

Measuring the designed survey's reliability is crucial to producing useful results in HCI research. Therefore, comprehensive survey designers must pay attention to these reliability measures while designing their surveys, ensuring that the results obtained from their surveys in current and future studies will have high reliability, thereby improving the theoretical and practical impact of their research.

Other survey evaluation issues. A large part of evaluation of the surveys usually happens after they are administered. However, in most cases an equally crucial evaluation of a survey happens just prior to the implementation. This evaluation is the actual *pilot testing* of the survey, sometimes also referred to as "pre-testing" (Dillman, 2000). As is the convention with most experimentation techniques in HCI research, the near-complete surveys can be administered to a group of participants. After this initial administration, the participants can be asked some questions about positive, negative, and missing issues in

the survey, and any questions or sections that were incomprehensible or unclear. Based on the feedback, the administrators can revise the survey and prepare it for the final administration. While there are no firm guidelines regarding the number of participants the pilot testing of surveys should be run on, in most cases a minimum of three participants is recommended for moderate-size surveys (less than 200 participants). For large-size surveys, up to 10 participants are generally useful (Dillman, 2000), although there is no upper limit for the number of participants to be used for the pilot study. Most surveys require revision after the pilot study, as in most cases there are some points the survey designers miss without the perspective of actual survey participants. In rare cases when no revisions are made to the surveys after the pilot survey administration, data obtained from the participants in the pilot can be included in the actual participant data pool.

How well a survey is designed is directly related to the validity and reliability of the results the research produces. Hence, the evaluation techniques covered in this section are crucial to the overall success of the designed survey and the research itself.

Survey Design, Redesign, and Revision

Design, redesign, and revision procedures for surveys to some extent bear some similarities to product design, redesign, and revision procedures. The initial design of surveys, as explained earlier, consists of generating questions based on the literature, expert opinion, and heuristics. Redesign and revision procedures mostly rely on the implementation of the survey on the entire group or a subgroup of participants. In most cases after a pilot test, surveys need revision which consists of changing or revising questions, scales, or instructions in the survey to make them clearer and more understandable for the participants. If there are errors in question structures or spelling errors, those are also located and eliminated after the pilot study. In rare cases, the required changes may be significant to the level that the survey may need redesign through the revision and change of most questions, scales, and instructions. It can be said that most small-scale survey revisions happen based on the feedback obtained from the pilot study.

Showing a certain amount of similarity to consumer products, frequently used surveys also need redesign and revisions over longer periods. Specifically in HCI, user habits and evaluation criteria for technology and technology-related products and issues change. It is therefore recommended that validated surveys that are used as empirical measurement tools in HCI should be reevaluated and updated about once a year to ensure they are up-to-date measurement tools and contain the latest additions to the HCI area as far as evaluation and opinion elements and techniques are concerned.

Illustrative Survey Design Example

Figure 58.3 presents a sample of a complete, generic paper-and-pencil survey in the example of a postexperimental task satisfaction survey. In the design of a survey measuring the Tablet PC usability issues among academic professionals, the first step is to develop a literature portfolio. This portfolio should cover literature on both mobile computer usability and Tablet PC usability. Next, researchers may send an inquiry to a group of academicians who use Tablet PCs, inquiring about major usability categories in relation to Tablet PCs in open-ended questions. Based on the input from literature and expert opinions, the researchers create an initial set of questions and scales, pilot-test it, and administer the survey, most likely in an environment where they give specific Tablet PC tasks to participants in a controlled environment, preceded by the actual survey administration.

Challenges in Survey Design

Survey design challenges mostly deal with possible mistakes in producing the survey questions and scales. Additionally, some problems may occur due to the questions having no validity backing. Therefore, the key for HCI survey researchers is to gain strong background in the area through literature and expert opinion before designing the surveys. After this background is gained, researchers are likely to have no difficulty designing the surveys with the appropriate number and content of questions and scales, and a comprehensive set of instructions to be presented to the participants on how they should fill out the survey.

SURVEY IMPLEMENTATION IN HCI

Survey implementation can be categorized into open and controlled survey implementation environments. In this section, these two environments are first discussed, followed by a discussion of sample representativeness issues, an implementation example, and discussions of implementation challenges and emerging and future trends in survey design and implementation.

Open vs. Controlled Implementation Environments

Survey implementation (also referred to as "survey administration") occurs in two alternative environments. In controlled survey implementation environments, participants fill out the survey in an environment specifically designated for and arranged according to their activity. An open implementation environment, on the other hand, does not contain any specific environmental traits for the participant to implement the survey. Open environments also mostly do not include any restrictions in time or other factors.

Controlled environments for implementation of surveys usually have the goal of preventing any distraction for the participant to hinder his or her understanding or judgment. Controlled survey environments are in most cases well-lit experimental rooms with appropriate equipment to make the participant moderately comfortable (often consisting of a chair and a table). Both computer and paper-and-pencil-based surveys can be implemented in either open or controlled environments. If the survey is implemented in a controlled environment in front of a computer, the survey implementers need to make sure that the computer's alignments (screen brightness, glare, screen distance, keyboard

GENERAL SATISFACTION SURVEY

Please indicate how you personally feel about performing the different tasks.
Each of the statements below is something that a person might say about performing a job like the searching task through an interface. You are to indicate your own personal feelings about your work experience with the different tasks you just completed by marking the number most closely describes how much you agree with each of the statements.

1. My opinion of myself went up when I performed the tasks correctly.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

2. Generally speaking, I am very satisfied with performing the tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

3. The tasks I performed were very meaningful to me.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

4. I felt the current web page structure design is good enough for me to perform the tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

5. The tasks were usually interesting enough to keep me from getting bored.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

6. My own feelings were not affected much one way or the other by how well I performed the different tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

7. I am in general satisfied with the kind of work I performed in the different web-page tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

8. Most of the things I had to do to perform the tasks seemed useless or trivial.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

9. I felt uncomfortable when I performed the tasks incorrectly.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

10. I felt very satisfied with the accomplishment I got from performing the tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

11. I felt very satisfied with the amount of independent thought and action I could exercise in the tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

12. With current web page structure design, I felt difficult to perform the tasks efficiently and effectively.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

13. I felt very satisfied with the amount of challenge in these tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

14. I felt very satisfied with the level of mental effort required to perform the tasks.

○	○	○	○	○	○	○
Strongly Disagree	Disagree	Moderately Disagree	Neutral	Moderately Agree	Agree	Strongly Agree

FIGURE 58.3. Sample of a paper-and-pencil satisfaction survey.

height, and other ergonomics issues) are optimized for the participant. In controlled environments, in most cases a survey administrator is also available to answer possible questions from the participant. These types of controlled survey administration environments are usually used to implement evaluation surveys as in most cases participants had just completed computer-based tasks and for them to be able to evaluate the interfaces or any other HCI-related environments, products, or procedures, controlled environments force them to do those evaluations immediately, while the memories of the items for them to evaluate are still fresh in their minds. Opinion surveys are generally not implemented in controlled environments. In some rare cases in which recording participant behavior during survey implementation is part of the experimentation, a controlled environment

can provide the equipment to nonintrusively record participant behavior/activities during the implementation of the survey.

While surveys are in some cases implemented in closed environments, doing so may be costly and time-consuming. Therefore, unless there is explicit need for a controlled environment, surveys are more commonly implemented in open environments. Open environments are environments of the participants' choosing, in most cases environments from their daily life. In open environments, a survey administrator is not present. Implementing surveys in open environments has the advantage of giving the participants the choice to choose the time and place of the implementation. This flexibility above all increases the ease of finding participants. Additionally, the freedom for the participants to fill out the survey at their convenience also improves their feeling of freedom and may increase their enthusiasm, thereby improving the accuracy of the survey results (Dillman, 2000). On the other hand, the surveys being filled out without the presence of a survey administrator will prevent the participants from asking any questions regarding the survey during the administration. Additionally, previous research has indicated that if participants have no particular motivation to fill out the survey, they may complete it very quickly without paying much attention, resulting in inaccurate survey results (Cochran, 1977).

Both controlled and open survey implementation environments have advantages and disadvantages. In most cases, however, open environments are faster and more convenient to collect the needed data due to the flexibility they offer to both the participants and implementers.

Sample Representativeness in Survey Implementation

To ensure the validity of the results obtained from surveys, it is imperative to choose a representative sample of the target population to successfully implement the survey. It is common knowledge that the validity of the survey results improves with larger sample sizes. Therefore, researchers need to carefully choose both the sample sizes and the sample participants.

There are no strict rules for determining sample sizes in survey implementation. The size of the survey sample, meaning how many participants should fill out the survey, depends on the type of the survey and survey questions, as well as the number of questions in the survey. Thiemann and Kraemer (1987) summarized the statistical methods of determining sample sizes based on the number of variables being measured in the experiment. In most cases, surveys measuring general topics (for example, surveys about cell-phone-use habits of a certain population, such as college students) should be implemented on relatively large sample sizes, possibly no less than 60 participants for a survey consisting of up to 30 questions. For survey implementation, as a rule of thumb the number of participants should always be bigger than the number of questions in the survey. While there are no set rules on choosing sample sizes for survey implementation, large sample sizes always improve the probability of obtaining high validity of surveys. For more on how to calculate optimal sample sizes in survey implementation to obtain satisfactory statistical power, see Thiemann and Kraemer (1987).

When surveys are administered, one of the most critical issues is to administer the survey on a balanced, nonbiased, and representative sample. In general, for surveys administered online, the survey should be sent to an as big a potential participant pool as possible to ensure heterogeneity. A large group of potential participants should also be sought if the surveys are paper-and-pencil-based, by, for example, mailing out a large number of paper surveys. A balanced sample size in terms of race and gender ensures higher validity as well as a broader application of the results, and this can be achieved by sending the survey to a large base of participants. The demographic information collected also helps determine exactly whether the participant sample accurately represents the target population. As in any experimental or survey-based research, heterogeneity of the sample size allows the researchers to strengthen their argument that the results of their study are applicable and generalizable to the majority of their target population. For example, if there is a vast majority of males compared to females in the sample while the gender distribution in the actual target population is estimated to be about even, then the validity of the results may be argued upon as the variation of the results due to the females would not be taken into consideration in the population, and consequently in the conclusions derived from the survey results. Therefore, survey participant pools should be chosen carefully, and they should also adequately represent the target population. Especially in large sample sizes, sample characteristics may vary greatly, especially concerning age, education, and occupation demographics. In these cases, the demographics should be presented in detail as part of research results. In most cases, a detailed explanation of demographics concerning participants' education and occupation can justify the representativeness issue of survey results, as long as the levels of these attributes do not differ very greatly between the sample and estimated target population. In those cases, the results of the survey findings should indicate that the findings are likely to apply to the particular segment of the population that had an overwhelming majority among participants in the sample size. For example, suppose a survey on habits of general population concerning the use of cell phones has been conducted. If the vast majority of the participants (more than about two thirds) are university students, then the researchers should indicate in their report of results that they measured the cell phone habits of the university student population, which constitutes to a large percentage of avid cell-phone users (Ozok & Wei, 2004).

Survey participant pools require caution when they are chosen, and in cases when the researchers are convinced that the sample is not highly representative of the target population, they need to make clear that the results of their survey may possibly have a narrower focus. In most cases, such narrowing of the target population which the research is aimed at does not result in validity problems, but rather makes clear which population or population segment the results of the survey study apply to.

Challenges in Survey Implementation

As indicated in previous sections, surveys are a relatively easy way of collecting data. They usually don't require expensive equipment for implementation, and with the advent of the Internet, can be easily distributed, filled out, returned, and analyzed. However, there are still some serious challenges in survey implementation in both paper-and-pencil and electronic environments.

Looking at the big picture, surveys are sometimes referred to as "subjective measurements." While the author of this chapter strongly disagrees with this statement, the distinction should be made between measurement environments where performance measurements are taken objectively and unequivocally through camera recordings and software, and environments where participants are asked to indicate what is going through their minds. In the latter environment, obviously, there is no way to ensure that participants are putting on the surveys exactly what they think about an issue, an environment, a tool, a product, or a procedure. It is not uncommon that participants fill out a survey without paying much attention, or even randomly mark responses without reading the questions. The survey-reliability measurement techniques to some extent prevent this type of random data from being used. For example, whether participants took the survey questions seriously can be determined by looking for discrepancies between Cronbach's Alpha duplicate questions. Additionally, strict instructions given to the participants at the beginning of the survey in written or spoken format can also to some extent improve the probability of participants taking the time to read the questions and give replies carefully. Researchers using surveys as their primary research tool always reserve the right to eliminate participant data that look ill fated or incomplete. However, the researchers need to have evidence in their hands that the participant did not complete the survey by obeying the rules that were presented to them, not on any other ground such as the participant responses not being in accordance with the majority of other participants or with the direction of results that are expected from the research.

Another challenge is the "return rate" of surveys. Response and return rates among surveys that do not offer any compensation is extremely low—less than 20% (Dillman, 2000). HCI research may involve lengthy surveys as well (more than 30 questions) which can potentially result in even lower return rates. Therefore, it is recommended that some sort of compensation should be offered to participants in survey research, whatever the resources will allow. This compensation may be small gifts (e.g., a pen or notepad), gift certificates, or cash compensation. Any of these types of incentives will surely improve the return rates of surveys (Miller, 1996). Additionally, surveys can contain statements to convince participants that they will also learn important issues concerning the research while filling out the surveys. It should be noted, however, that the practical issue of finding subjects should not bias the sample, and recruiting of participants should be arranged according to the data-collection needs of the research, not according to what kind of participant groups are the most practical to collect data from. Recruitment activities need to be targeted to ensure a representative sample.

In most implementation activities, participants should be given their privacy while filling out the survey, ensured that their data will be kept confidential, and be provided a comfortable environment. Otherwise, they may want to either quit or finish as soon as possible without any consideration of the accuracy of responses. It should also be noted that surveys are voluntary, and therefore survey implementers should indicate the voluntary nature of the surveys and not pressure the participants. Research indicates that when participants are pressurized to give accurate responses or when a mutual trust between the administrator and the participant is not established regarding the sincerity of both sides, they mostly produce very unreliable survey results (Dillman, 2000).

Other survey implementation challenges involve participants' interaction with the survey interface. In paper-and-pencil surveys, the fonts on the paper should have enough size and familiarity for all participants, a pencil with an eraser head will allow participants to correct their responses, and survey elements such as questions and scales should be adequately distinct from each other to prevent any mistakes. Survey designers should use a very simple language in the surveys and avoid any little-known words and sentence structures (Gendall, 1998). Surveys consisting of multiple pages should be clearly numbered on each page. This kind of a convenient interaction environment will improve participant enthusiasm and increase response rates as well as accuracy of survey results. Additionally, while some studies expressed concern about differences in survey responses among computer and paper-and-pencil surveys (Sanchez, 1992; Cole, 2005) and issues concerning the format in which online surveys are presented (Couper, Traugott, & Lamias, 2001; Couper, Traugott, Conrad, & Crawford, 2004; Kiernan, Kiernan, Oyler, & Gilles, 2005), a recent study indicated that the accuracy of survey responses did not significantly differ between surveys administered online and those administered paper-and-pencil (Carini, Hayek, Kuh, Kennedy, & Ouimet, 2003).

In electronic surveys, the computer interface should be very simple and participants with little experience with computers should not have any difficulty using the interfaces. In most cases, participants are required to mark their responses with a mouse and type their responses on clearly marked, designated text spaces that have adequate size for visibility purposes. Some special equipment may offer some additional conveniences, such as touch screens. Screen glare and font sizes should be given consideration to. It can be recommended that computer surveys should be implemented on screens no smaller than 12 inches of diagonal size, with a refresh rate of at least 60 MHz. For surveys longer than one screen, scrolling should be minimized. It is recommended that each survey screen should not require more than two screen-heights of scrolling, and should be connected with hyperlinks or screen buttons, meaning once a participant completed a screen, he or she should be required to move on to the next screen by clicking on a screen button or a link. Besides the scrolling issue, if a participant sees a lengthy survey all presented on one screen, he or she may get discouraged to fill out the survey due to its length.

An additional potentially problematic item is the number of questions to ask participants in a survey. In most cases, the attention span of participants is very short, and surveys that do not offer any compensation are recommended to be shorter than 30 questions. In most cases, participants are not interested in spending more than 15 minutes in filling out surveys for which they don't get any compensation. There is always a tradeoff between the size of surveys, meaning the ability to collect all the necessary data, and the ability to recruit subjects. Long surveys are more difficult to recruit participants for. Researchers should think carefully about compensation methods (money, gifts, gift certificates) if they intend to implement large-scale surveys.

Finally, Internet surveys also carry the potential of technical difficulties due to the variety of computers the participants may be using. Schleyer and Forrest (2000) identified usability problems,

programming errors, and incompatibilities/technical problems as main problems identified in web-administered surveys. Therefore, Internet-based, especially web-based surveys should not require any scripts or plug-ins to run and if possible should consist of simple Hypertext Markup Language (HTML) code, as HTML is universally recognized by all browsers.

These are the major challenges currently the survey implementers have to deal with. However, with careful design and implementation, as well as strict instructions containing comprehensive information presented to the participants regarding the survey, the challenges can be easily overcome, resulting in valid and reliable survey results.

EMERGING AND FUTURE TRENDS IN SURVEY DESIGN AND IMPLEMENTATION IN HCI

Surveys have retained their structure, more or less, for many decades. It can be said, however, that electronic and especially Internet-based implementation has changed the convenience level of survey implementation and data analysis in a significantly positive way. It is difficult to predict whether any new groundbreaking techniques will cause further leaps in survey design, development, and analysis, but if significant new developments will happen in the near future, they are likely to happen in the implementation technology. Internet surveys are on the rise, with the percentage of Internet-based surveys being on the rise for the past five years. While Internet-based surveys comprised 15% of all surveys implemented in 1999, this number increased to 70% in 2004, according to Nua Internet Surveys (2005). With the improvement in voice recognition and voice synthesis technologies, future surveys may eliminate the need of a visual interface; however, human-factors issues in these types of interfaces specifically for surveys are still to be explored. It is apparent that the number of surveys implemented through the Internet and other networks will continue to climb in the years to come, due to cost savings and a number of other convenience issues. In the future, HCI research is also likely to continue to use surveys as a main data-collection tool. With HCI research becoming a more integral part of technology design and creation (for example, human-factors engineers and software engineers working collaboratively in the software-design process), user surveys may become more integrated, collecting data for current or future users regarding both HCI and other technology issues. Additionally, computer literacy is increasing at a fast pace (U.S. Census Bureau, 2005), which allows HCI survey researchers more freedom in asking more sophisticated questions concerning interface evaluation, current user trends, and more. All that said, the contribution of surveys to HCI research is highly significant these days, and will likely continue to be so in the many years to come.

References

Aiken, Lewis R., & Lewis A. (1996). *Rating scales and checklists: Evaluating behavior, personality, and attitudes* (1st ed.). NY: John Wiley & Sons.

Badre, A. (2002). *Shaping web usability: Interaction design in context*. Boston, MA: Addison Wesley Professional.

Baecker, R., Grudin, J., Buxton, W., & Greenberg, S. (1995). A historical and intellectual perspective. In Baecker, R. M., Grudin, J., Buxton, W. A. S., & Greenberg, S. (Eds.), *Readings in human-computer interaction: Toward the year 2000* (2nd ed.) (pp. 35–47). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Baker, R. (1998). The CASIC future. In M. P. Couper (Ed.), *Computer assisted survey information collection* (1st ed.) (pp. 583–604). New York, NY: John Wiley & Sons.

Belson, W. (1981). *The design and understanding of survey questions*. Aldershot, England: Gower.

Card, S. (1996). Pioneers and settlers: Methods used in successful user interface design. In M. Rudisill, et al. (Eds.), *Human-computer interface design: Success stories, emerging methods, and real-world context* (pp. 122–169). San Francisco, CA: Morgan Kaufmann Publishers.

Carini, R. M., Hayek, J. H., Kuh, G. D., Kennedy, J. M., & Ouimet, J. A. (2003). College student responses to web and paper surveys: Does mode matter? *Research in Higher Education, 44*, 1–19.

Cochran, W. (1977). *Sampling techniques* (3rd ed.). New York, NY: John Wiley & Sons.

Cole, S. T. (2005). Comparing mail and web-based survey distribution methods: Results of surveys to leisure travel retailers. *Journal of Travel Research, 43*(4), 422–430.

Converse, J. M., & Presser, S. (1986). *Survey questions: Handcrafting the standardized questionnaire*. Newbury Park, CA: Sage Publications, Inc.

Couper, M. P., Traugott, M. W., Conrad, & Crawford (2004). What they see is what we get:Response options for web surveys. *Social Science Computer Review, 22*(1), 111–127.

Couper, M. P., Traugott, M. W., & Lamias, M. J. (2001). Web survey design and administration. *Public Opinion Quarterly, 65*, 230–253.

Cronbach, L. J. (1990). *Essentials of psychological testing*. New York, NY: Harper & Row Publishing.

Dillman, D. (1991). The design and administration of mail surveys. *Annual Review of Sociology, 17*, 225–249.

Dillman, D. (1999): Mail and other self-administered surveys in the 21st century. *The Gallup Research Journal, Winter/Spring 1999*, 121–140.

Dillman, D. (2000). *Mail and Internet surveys: The tailored design method.* New York, NY: John Wiley and Sons, Inc.

Gendall, P. (1998). A framework for questionnaire design: Labaw revisited. *Marketing Bulletin, 9*, 28–39.

Hackman, R. J., & Oldham, G. R. (1980). Development of job diagnostic survey. *Journal of Applied Psychology, 60*(2), 159–170.

Jenkins, C. R., & Dillman, D. A. (1997). Towards a theory of self-administered questionnaire design. In L. Lyberg, P. Biemer, M. Collins, E. deLeeuw, C. Dippo, N. Schwarz, & D. Trewin (Eds.), *Survey measurement and process quality* (pp. 165–196). New York, NY: John Wiley & Sons.

Kiernan, N. E., Kiernan, M., Oyler, M. A., & Gilles, C. (2005). Is a web survey as effective as a mail survey? A field experiment among computer users. *American Journal of Evaluation, 26*(2), 245–252.

Krosnick, J. A. (1999). Survey research. *Annual Review of Psychology, 50*, 537–567.

Litwin, M. (1995). *How to measure survey reliability and validity.* Thousand Oaks, CA: Sage Publications.

Lund, E., & Gram, I. T. (1998). Response rate according to title and length of questionnaire. *Scandinavian Journal of Public Health, 26*(2), 154–160.

Medsker, G. J., & Campion, M. A. (1997). Job and team design. In G. Salvendy (Ed), *Handbook of Human Factors and Ergonomics* (3rd ed.) (pp. 450–489). New York: Wiley.

Miller, K. (1996). *The influence of difference techniques on response rates and nonresponse error in mail surveys.* Unpublished Master's thesis, Western Washington University, Ballingham, WA.

Myers, B. (1996). *A quick history of human computer interaction.* Carnegie Mellon University School of Computer Science Technical Report. CMU- S-96-163 and Human Computer Interaction Institute Technical Report. CMU-HCII-96-103, August, 1996.

Myers, B., Hollan, J., & Cruz, I. (1996). Strategic directions in human computer interaction. *ACM Computing Surveys, 28*(4), 794–809.

Nielsen, J. (1989). Coordinating user interface for consistency. *SIGCHI Bulletin, 20*, 63–65.

Nielsen, J. (1993). *Usability engineering.* London, UK: Academic Press.

Nua Internet Surveys (2005): nua.com, site accessed 9.15.2005.

Ozok, A. A., & Salvendy, G. (2000). Measuring consistency of web page design and its effects on performance and satisfaction. *Ergonomics, 43*(4), 443–460.

Ozok, A. A., & Salvendy, G. (2001). How consistent is your web design? *Behaviour and Information Technology, 20*(6), 433–447.

Ozok, A. A., & Wei, J. (2004). User perspectives of mobile and electronic commerce with a usability emphasis. *Proceedings of the ISOneWorld 2004 Conference*, Las Vegas, NE, Article 71.

Sanchez, M. E. (1992). Effect of questionnaire design on the quality of survey data. *Public Opinion Quarterly, 56*, 206–217.

Schleyer, T. K. L., & Forrest, J. (2000). Methods for the design and administration of web-based surveys. *Journal of American Medical Informatics Association, 7*(4), 416–425.

Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction.* New York: Addison-Wesley.

Stone, D. H. (1993). Design a questionnaire. *British Medical Journal, 307*(6914), 1264–1266.

Thiemann, S., & Kraemer, H. C. (1987). *How many subjects?: Statistical power analysis in research.* Newbury Park, CA: Sage Publishing.

U.S. Census Bureau (2005). Computer and Internet use in the United States: 2003. *Annual Report Special Studies*, 23–108.

Zhang, Y. (2000). Using the Internet for survey research: A case study. *Journal of American Society for Information Science, 51*(1), 57–68.

·59·

INSPECTION-BASED EVALUATIONS

Gilbert Cockton and Alan Woolrych
University of Sunderland

Darryn Lavery
Microsoft Corporation

Usability Inspection Methods **1172**
Definitions 1172
The Need and Rationale for UIMs 1172
The Logical Structure of Usability Inspection 1172
Analyst Preparation 1173
Candidate Problem Discovery 1173
Confirmation and Elimination of Candidate Problems 1174
Six Example UIMs **1175**
Heuristic Evaluation 1175
Cognitive Walkthrough 1175
A Hybrid Method: Heuristic Walkthrough 1176
Other UIMs 1176
Ergonomic Criteria 1176
Cognitive Dimensions 1176
Pluralistic Walkthrough 1177
Assessing and Scoping Methods **1177**
Thoroughness, Validity, and Effectiveness 1177
Other Quantitative Measures for UIMs 1178
Qualitative Analysis of UIM Usage 1179
Methodological Issues in UIM Assessment 1179
Statistical Conclusion Validity 1179
Internal Validity 1179
Construct Validity 1179
External Validity 1180
Conclusion Validity 1180
Improving UIM Assessment and Comparison 1180
Analyst Training and Preparation 1180
Reliability of Prediction Merging 1180
Ability of User Testing to Expose Actual Problems 1181
Reliability of Known Problem Extraction 1181
Reliability of Matching Predicted to Actual Problems and Structured Report Formats 1181
Controlling Confounding Variables 1181
Outstanding Methodological Challenges 1182
Understanding Inspection Methods **1182**
The DARe Model of UIM Effectiveness 1182
The DARe Model and the Scoping of UIMs 1182
The DARe Model and the Development Lifecycle 1184
Improving Inspection Methods **1184**
Method-Independent Improvements 1185
Method Specific Improvements 1185
Improving Analysts 1186
Future Trends and Best Practice **1187**
Prospects for UIMs 1187
Best Practice for UIMs 1187
Note **1187**
References **1188**

USABILITY INSPECTION METHODS

Usability inspection methods (UIMs) remain an important discount method for usability evaluation. They can be applied to any designed artifact during development: a paper prototype, a storyboard, a working prototype (e.g., in Macromedia Flash™ or in Microsoft PowerPoint™), tested production software, or an installed public release. They are analytical evaluation methods, which involve no typical end users, unlike empirical methods such as user testing. UIMs only require availability of a designed artifact and trained analysts. Thus, evaluation is possible with low resources (hence discount methods). Although risks arise from low resources, well-informed practices disproportionately improve analyst performance, improving cost-benefit ratios. This chapter introduces UIMs, covering six and one further method, and provides approaches to assessing existing, emerging, and future UIMs and their effective uses.

Definitions

A UIM is an analytic evaluation method that is applied directly to an interaction design artifact. Direct application contrasts UIMs with model-based methods, which are indirectly applied via design representations (or models), requiring construction of models, and secondary application of analyses to designed artifacts. Using the GOMS method, for example, a task model would be analyzed, and the results would have to be reframed to address the actual design artifact. In contrast, UIMs such as heuristic evaluation (discussed in a later section) directly identify design features that may cause user difficulties.

The Need and Rationale for UIMs

All evaluation methods aim to find relevant usability problems and to support understanding and fixing them. Systems contain potential causes of user difficulties that are only exposed within certain usage scenarios. Analytical methods must simulate (fragments of) such scenarios. User-based methods remain more reliable because, once selected and briefed, individual test participants generate many complete usage scenarios. Hence, user testing can expose problems through user behaviors that are very hard to anticipate. However, testing only reveals a wide possible range of usability problems if users stress the system with complex interactions across all features. It is thus possible to anticipate user difficulties that may not emerge during testing, unless specific efforts are made to flush out all predicted problems (Woolrych, Cockton, & Hindmarch, 2004).

There is thus no absolute effectiveness or quality ranking of analytical and empirical methods. Where resources allow, one should combine both approaches; where resources are limited, it may only be possible to apply UIMs. Thus, heuristic evaluation was the most used usability method in a paper questionnaire-based survey of 111 usability professionals in 1999 (Rosenbaum, Rohn, & Humburg, 2000). However, a more recent web survey of 83 usability professionals (Venturi, 2004) now ranks inspection methods (38% of respondents use during design) well below discount user testing (quick and dirty usability tests used by 53%—in contrast to 70% and 65% respectively in the 1999 survey). Although surveys must be compared with great caution, inspection methods may no longer be the predominant method in usability practice.

A possible shift from analytical to empirical methods may be due to excessive pessimism about UIMs' weaknesses relative to other methods. Our view is that, used with care, inspection methods remain a key technique in the usability toolbox: "The challenge is to improve all HCI methods, so that discount methods are less discounted and 'full strength' methods can be applied in more contexts" (Cockton & Woolrych, 2002, p. 29).

The most appropriate use of inspection methods is to drive design iterations, rather than use for summative evaluation, benchmarking, or competitor analysis. UIMs are cheap to apply and are seen as low cost and low skill. They have been applied to many commercial designs by practitioners. UIMs can be used before a testable prototype has been implemented and can be iterated without exhausting or biasing a group of test participants. UIMs can also be a planning resource for user testing, which can be designed to focus on predicted problems. Also, the value of UIMs as training devices should not be underestimated. In contrast, model-based approaches remain limited or immature, are expensive to apply, and use is mostly restricted to research teams (e.g., Salvucci, Zuber, Beregovaia, & Markley, 2005). Similarly, development resources may rule out user testing, leaving UIMs as the only possible approach, and some usability is always better then no usability.

UIMs remain important and their weaknesses can be mitigated. The main risks are missing serious usability problems, and wasting development effort through poor fixes to mispredicted nonproblems (false positives). Different business models make different demands, thus UIM errors are more costly in some development contexts than others are. In some contexts, successful predictions may always be worthwhile despite a flood of false positives, as fixed problems translate into savings on support costs and attractive new features. When users buy software, most must struggle on, unlike visitors to free e-commerce sites, yet mainstream UIMs do not cover their complete user experience especially for affective issues such as trust, comfort, and brand image. Hence, user testing is vital to eliminate any severe problems; otherwise, money will be lost.

The Logical Structure of Usability Inspection

Well-informed usage of UIMs is essential to their effective applications. This can be based on an understanding of UIM usage, of where methods can fail, and how this can be avoided. A high-level structure, which can be readily inferred for usability inspection, supports such an understanding. It is a logical rather than a temporal structure. Its phases are logically distinct but overlap during inspection. Inspection is essentially a search problem. Analyst behavior is close to the generate and test strategy from early artificial intelligence research, where possibilities are first generated (e.g., moves in a chess game) and then tested to identify possibilities with specific attributes (e.g., the best next move). Note that the role of evaluation does not extend to fixing problems, even though methods such as RITE (Medlock, Wixon, McGee, & Welsh, 2005) do place fixing problems within scope. RITE is structured around three questions: Is it a prob-

lem? Do we understand it? and Can we fix it? Only the first question and some of the second are within scope of evaluation methods. The rest of RITE (really understanding and fixing problems) is an iteration method with (empirically) grounded redesign decisions, based on all relevant project information.

The first step is for analysts to study a UIM. Then, each time a UIM is applied, analysts must study the target design in the form provided. These preparation steps are followed by two logical generate and test phases of UIM application. We can thus model usability inspection as having three distinct logical phases: (a) analyst preparation, (b) candidate problem discovery, and (c) confirmation or elimination of candidate problems. We now review each logical phase.

Analyst Preparation

Analysts must first understand a UIM (and continually revisit this understanding). They must then understand a design and its expected usage context. Gaps in any understandings of the UIM and a design translate into poor predictions (and thus flawed design change recommendations).

UIMs are still developing and have yet to mature. Analysts are thus always researchers who have much to contribute to UIM development. The oldest UIM, guidelines conformance, is over 30 years old, but no guideline set has survived (unchanged). The most structured UIM, cognitive walkthrough, is much younger and changed often in its first four years, but has been largely undeveloped since then, with some exceptions (e.g., Spencer, 2000).

Such short and turbulent histories give analysts little solid to draw on when learning UIMs. Methods tend to be reported in short conference papers, with the occasional journal paper or a chapter in a student text or practitioner's book. Unpublished tutorial notes are often the best sources. In reality, evaluators must consult a small set of sources and then work out how to apply a UIM, taking into account any available assessments.

Novice analysts should practice UIMs on familiar systems and usage contexts to establish quickly their scopes and accuracies with a UIM. Once analysts are confident that they understand a UIM, they can then apply it, but they must be well briefed about designs and their intended usage contexts. Without knowledge of the latter, analysts must guess how and when potential problems would really be severe or find features and behaviors that are so incompatible with general human capabilities that usage contexts become irrelevant. Not surprisingly, relatively few developers are so unaware of human limitations that such extreme features get introduced—for example, the authors have only once encountered pale lemon text on a white background. Typical usability problems tend to be more subtle and closely related to contextually specific factors, such as user knowledge (e.g., of similar systems), semiotic expectations (e.g., expected meanings of system elements), domain factors (e.g., number formats for accounting), human-activity structures (e.g., you do not pay before you buy), and resources (e.g., there simply is not time available for a task like that). Such knowledge of expected usage contexts lets analysts make sound judgments about a problem's likely severity, persistence, or frequency.

Unfortunately, no UIM integrates contextual information and its understandings of users and tasks according to Cockton and Woolrych. (2002). Fortunately, HCI professionals apply contextual research (Venturi, 2004), even though some may not accept that contextual methods remain separate from UIMs (Manning, 2002). Not every UIM user will apply them unprompted. UIM expertise will thus largely depend on analysts' existing skills and knowledge, unless extended by tutorial material with self-assessment, as in our studies (Lavery, Cockton, & Atkinson, 1996a, 1996b, 1996c; Lavery & Cockton, 1997a).

Lastly, an analyst must know about and understand the system they are inspecting. Some UIM developers disagree, preferring to keep analysts in untainted ignorance. They aim to induce user empathy, but this can only be truly grounded in contextual research. Ungrounded beliefs about users and usage contexts can be very unreliable (and often insult real users' intelligence). If analysts begin inspection ignorant of both the usage context and the system being evaluated, then the result is not better empathy with the user but incorrect claims that features are missing or beyond users. While there may be evidence of usability problems in such analyst errors, they are still errors. Properly informed analysts are more likely to note that a feature is hard to find (rather than absent) or that a design rationale has overlooked some contextual criteria. Such problem predictions are far preferable to bogus ones, which are major risks in UIM usage.

Candidate Problem Discovery

Analysts will only be well placed to find possible problems if they adequately understand a UIM, a system and its application domain, its design rationales, and its expected context of use. They can then use a UIM to discover possible usability problems. UIM usage takes two distinct forms. The predominant one is rule-based conformance assessment, where a usable system will adhere to a set of design rules. The other is the walkthrough, where a usable system is one where no likely user difficulties emerge during step-by-step analysis of hypothetical interactions.

Rule-based methods operate at different levels of abstraction. The most abstract detect breaches of principles that express properties of systems and interaction. Often known as "-ities," such principles include browsability, predictability, and flexibility. The least abstract methods look for breaches of style rules that prescribe design decisions at a level of details such as the structure and content of window title bars. In between principles and style guides are heuristics (less abstract than principles) followed by guidelines (more abstract than style guides are).

Heuristics and guidelines can be expressed with varying specificity over a range of abstraction. Heuristics as rules of thumb tend to be general, but can be specific (e.g., provide clearly marked exits). Guidelines range from specific design instructions to general wisdom (e.g., know the user).

Fig. 59.1 relates the five main UIM types to each other in terms of level of abstraction (vertical axis, top is high) and focus (horizontal axis, middle is interaction centered, left is system centered, right is user centered). The range of abstraction (height) mostly determines the coverage (area) for each method type, but it can also be extended by range of focus (width). Letters represent UIM types: P (principles), H (heuristics), G (guidelines), S (style), R (rules), and W (walkthrough). The rule-based

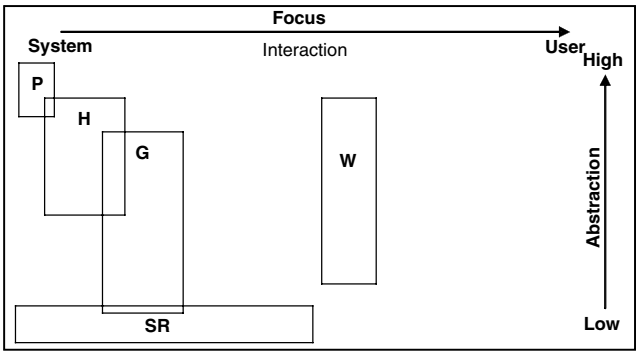


FIGURE 59.1. Relationship of types of inspection-based method by level of abstraction and focus.

types overlap with heuristics spanning abstraction levels within principles and guidelines, and guidelines overlap with style rules. Heuristics, guidelines, and style rules can all be specialized for particular application domains, and thus they span from almost wholly system centered towards a balance of system and interaction centered.

The level and range of abstraction of UIMs create different generic discovery tasks for analysts. More abstract rules require evaluators to narrow rules down to an instance, as well as the opposite, generalizing from an instance to a rule to detect breaches. Generalizing was performed more readily and confidently in a pilot study on problem matching (Lavery, Cockton, & Atkinson, 1997). However, generalizing search tasks must proceed feature by feature, scanning (all) rules for one that a feature breaks. In contrast, narrowing discovery tasks proceed rule by rule, scanning (all) features to see if any break them. Decision making thus becomes quicker as rules become less abstract, since less translation from rules to features is needed. However, more detail increases rule sets, enlarging the search space: not only must analysts search system features, but they must also search large rule sets for rules that features could breach.

If the preceding is a little daunting and confusing, then consider usability analysts. Clearly, parallel systematic searches of large system and method spaces are cognitively impossible, and thus analysts must develop other discovery methods to isolate possible problems. Ironically, analysts using usability heuristics must develop further discovery heuristics to apply them. The quality and application of discovery heuristics (or methods) determines whether analysts will find or miss possible usability problems. The latter is major source of risk in UIM usage, along with misprediction of bogus problems.

Confirmation and Elimination of Candidate Problems

Analytical methods arrive at conclusions by calculation, deduction, or judgment (and unfortunately wild guesses and imagination). Having discovered a possible usability problem, analysis is required either to confirm it as probable or to eliminate it as improbable. Empirical methods arrive at conclusions through observation, analysis, and inference. Empirical evaluations do not rely on predictive theories but simply see what happens during usage. Analytical methods thus rely critically on the quality of their underlying theories of HCI, which are explicit for model-based methods but tacit for inspection-based methods.

Just as the quality of discovery methods determines whether analysts will find or miss possible usability problems, so the quality and application of analysis resources determines whether analysts will correctly confirm or eliminate them. For analysis, knowledge of expected contexts of use, human capabilities, and key properties of HCI are critical. Contextual information may eliminate a possible problem as improbable. Knowledge of display-based interaction may eliminate possible problems that overlook users' abilities to discover information and to explore interactive behaviors. Knowledge of human capabilities such as visual attention may either confirm possible problems (e.g., key information in the wrong place) or eliminate them (e.g., misleading information in the wrong place).

Analysts need strong understandings of HCI in general, and of usage contexts in particular, to counterbalance simple deterministic underlying models of problem causation in rule-based methods. These are the main source of risk in UIM usage, leading to missed problems during discovery and false predictions during analysis. The overwhelming tendency is for rules to be applied to system features and not to the combined use of several features by real users in real interactions. Rule-based methods thus tend to be system-centric, incorrectly assuming that usability can be an inherent property of the system alone. This introduces the risk of both false positives (by overestimating features' impacts on users with average and above cognition) and false negatives (by failing to spot adverse interactions between usage and guideline-conformant features). Rule-based methods thus best identify problems at the extremes of cognitive and perceptual overload, which would require exceptional (and highly improbable) usage contexts to ameliorate them. For white text on a lemon background, such a context would involve some sort of custom optical filter to increase contrast (or a screen reader).

In contrast, procedural inspection methods prompt consideration of interaction beyond brief encounters with individual system features. In principle, they can bring any aspect of a usage context into consideration during walkthrough, and across a range of abstraction levels, ranging from a user's motives for fulfilling a role (that requires them to carry out a task) to the human information processing of keystroke level interaction. However, as with rule-based methods, increasing the set of concerns addressed at each walkthrough also increases the chances of error (assuming the method is practical enough for any use at all to be attempted).

When analyzing possible problems, the evaluator is again potentially overwhelmed. Again, analysts need to develop their analysis methods to make inspection manageable. While discovery methods determine whether a possible problem will be found or not, it is analysis methods that determine four inspection outcomes of (a) true positive (hit), (b) true negative, (c) false positive, and (d) false negative (a lost rather than a missed problem). Whereas in signal detection, the assumption is that a positive indication is always present, in usability inspection, analysts have to make their own signals during preparation and discovery, and thus a fifth inspection outcome of genuine miss is added. Only this inspection error can be attributed to failures during candidate problem discovery.

The above logical phase model of usability inspection will now be applied to six example usability inspection methods, which are first described and then analyzed.

SIX EXAMPLE UIMS

Heuristic Evaluation

Using a heuristic evaluation (Nielsen & Molich, 1990; Nielsen, 1992, 1994b), the evaluator uses a small set of rules of thumb to confirm usability problems by considering possible violations. For example, systems that require the user to type in the name of a file would breach an error prevention heuristic, while a system that allowed the user to select the name would not. There are documented accounts in the literature of applications of heuristic evaluation to interactive systems such as finance, travel, and office systems.

The heuristics in Fig. 59.2 were derived from seven sets of heuristics and guidelines, including an earlier version of Nielsen's (1994a) own, by analyzing how well they explained an existing database of 249 problems from usability evaluations of 11 different interactive systems.

For each usability problem, Nielsen (1994a) assigned a rating of one to five to each individual heuristic according to how well it explained the usability problem. Average ratings for each heuristic were then calculated. A principal components analysis was used to determine if a few factors accounted for most variability of the usability problems. Nielsen found that seven factors could only account for 30% of the variability, and 53 factors were needed to account for 90%, which is too many for a heuristic evaluation (HE). The names of the seven factors, which were chosen by Nielsen included "visibility of system status," "match between system and real world," and "error prevention."

Nielsen (1994a) attempted to pick the individual heuristics that provide the best explanation of the usability problems. A list of top heuristics was chosen by first picking the heuristic that found the most usability problems with a rating of three or more ("explains a major part of the problem, but there are some aspects of the problem that are not explained"), and then repeating with the remaining usability problems to complete the list. This procedure was also repeated to provide the best explanation of the more serious usability problems.

1. Visibility of System Status
2. Match between system and the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose and recover from errors
10. Help and documentation

FIGURE 59.2. Heuristics (Nielsen, 1994b).

Nielsen (1994a) found that a small number of heuristics could mostly explain both the serious and unfiltered problem set, and that in general the same heuristics appeared in both lists either as the exact same guideline or as similar guidelines. Furthermore, he found that almost all of the seven principle factors were covered in the top heuristics list. This supported his claim that the factors (the first seven heuristics in Fig. 59.2 along with eight and nine) form a basis for a set of heuristics that has remained the main resource for HE; the 10th, "Help and Documentation," was later added. It is not clear, however, how much the additional three heuristics extend the coverage of the sample problem set beyond 30%, but this is very unlikely to reach 40%.

However, Nielsen (1994a) noted that the new set is good at explaining problems, but he did not know how effective they would be at finding usability problems. Thus, the current version of HE comes with known limitations in its discovery resources, and that others have attempted to fill the gap is no surprise. Thus, Muller et al. (1995) rightly claimed that the 10 1994 heuristics ignore usage contexts. They added three extra heuristics to assess how well a design fits user needs and the work environment:

1. Respect the user and his or her skills
2. Pleasurable experience with the system
3. Support quality work

The value of these extra heuristics was demonstrated for an application where 247 usability problems were found, which led to 89 redesign recommendations of which 72% were implemented. Each problem and recommendation was scored for its connection to each of the 13 heuristics. The three new heuristics were solely responsible for identifying 15% of the problems and 10% of the recommendations. By rating problems and recommendations using a severity scale (from one to five), they found that problems revealed with the new heuristics were rated as being slightly less important than those identified using the 1994 heuristics, but there was no difference in importance for the recommendations. However, such recommendations are not solely the result of a UIM and the lack of a difference here may reflect the common redesign resources in use.

Cognitive Walkthrough

Cognitive walkthrough (CW; Wharton, Rieman, Lewis, & Polson, 1994; Lewis & Wharton, 1997) is a procedural inspection method. It assesses the learnability of walk up and use systems. One or more evaluators choose a set of tasks for analysis and document the assumptions about the target users of the design. Analysts step through each stage of the task answering a set of questions (see Figure 3) and backing up their answers with empirical evidence, experience, or scientific evidence. A negative answer to any of the questions indicates potential usability problems with the system. However, these have to be considered together in the context of the task and may result in either a success case (trivial usability problems) or a failure case (important usability problems).

CW is underpinned by CE+, a theory of learning (Polson & Lewis, 1990). For example, the theory predicts that a user new to a system will choose an action with a good match to the

1. Will the user try to achieve the right effect?
2. Will the user notice that the correct action is available?
3. Will the user associate the correct action with the effect that the user is trying to achieve?
4. If the correct action is performed, will the user see that progress is being made towards solution of the task?

FIGURE 59.3. Cognitive walkthrough questions (Wharton et al., 1994).

user's current goal. This reflects the relation between interaction centeredness and grounding in theory, in that the most system-centered methods are generally atheoretical.

Two main forms of CW have been presented in the literature. Earlier versions explicitly considered users' goals (e.g., Lewis, Polson, Wharton, & Rieman, 1990; Polson, Lewis, Rieman, & Wharton, 1992). For example, Polson et al.'s (1992) version required three pages of questions for each action; one page to address users' goals, one to cover choosing and executing the correct action, and a final page of questions covering the effect of taking the action on the user's goal structure. The overhead of the method and difficulty applying the method led to the cognitive jogthrough (Rowley & Rhoades, 1992) and multimedia tool support (Rieman et al., 1991) before a simplification of the method (Wharton et al., 1994; Lewis & Wharton, 1997).

Overall, first versions of CW provided extensive analyst preparation materials, but these proved to be unwieldy as discovery resources, reducing the benefits of being interaction centered. Still, the success case in CW is a rare example of an analysis resource for problem elimination, letting analysts argue that some apparent design flaws would not automatically cause severe usability problems. However, there is no method support in CW for forming success and failure cases.

In response to criticisms, Wharton et al. (1994) developed a simpler version of CW. This version "de-emphasizes the explicit consideration of the user's goal structure," resulting in the current set of four questions (see Figure 3), although a subsequent variation (Spencer, 2000) reduced that down to two. Overall, later versions of CW reduced extensive preparation materials for analysts, making them less unwieldy as discovery resources, but perhaps losing some benefits that arise with interaction-centered methods, apart from ones preserved via task descriptions.

A Hybrid Method: Heuristic Walkthrough

Heuristic walkthrough (HW; Sears, 1997) is a cross between HE and CW. The input to the method is a prioritized list of user tasks. These tasks should include frequent or critical tasks, but may also include tasks designed purely to ensure coverage of the system. HW thus provides some guidance on preparation, which in turn provides discovery resources.

There are two phases to the evaluation: a task-based phase and a free-form phase. In the first phase, the evaluators explore the tasks using a set of thought provoking questions derived from CW. They are free to explore the tasks in any order, spending as long as they need, but they should be guided by task priorities. In the second phase, evaluators are free to explore the system. They use the set of thought of provoking questions again, plus heuristics from HE.

Sears (1997) compared HW against CW and HE. HW had more actual problems than CW and had fewer false positives than HE. Overall, Sears' merge of HE and CW improved analyst preparation and discovery resources, with evidence of improved analysis resources (reduction in false positives), which may be due to the initial CW inhibiting discovery of unlikely problems. In Fig. 59.1, HW would span and bridge the areas covered by heuristics on the left and walkthroughs in the middle (at least some things in UIM research *are* simple).

Other UIMs

Ergonomic Criteria

Like HE, ergonomic criteria (Bastien & Scapin, 1995; Scapin & Bastien, 1997) uses a set of principles. However, these 18 ergonomic criteria (principles) are structured around a definition, a rationale, and examples of guidelines plus comments that include help to disambiguate any related criteria. They were formed by reviewing existing guidelines and experimental results (Scapin, 1990). The extra structure present in the ergonomic criteria is likely to assist analysts preparing to use this method.

Cognitive Dimensions

Green (1991) claimed that HCI evaluation techniques tend to focus too much on surface features, analysis tends to be over technical, and HCI has no common vocabulary. Cognitive dimensions (Green & Petre, 1996) attempts to capture a set of orthogonal dimensions for restricted set of systems called "notations," which are used to design information structures, for example programming languages and spreadsheets. These dimensions are not necessarily criteria for a design but a characterization of the design space.

A sample of dimensions can be found in Fig. 59.4. The dimensions embody a theory about how people use notations to build information structures. A simplified theory of action would involve people translating goals into specifications, then into actions, and executing them. Rather, people work bottom up, top down—goals and subgoals may be attacked at any moment. Green (1991) claimed the preferred strategy is oppor-

Hidden Dependencies can occur when one cannot see how a value is calculated, for example a spreadsheet where a small change can have unexpected effects. All dependencies that are important to the user must be accessible.

Premature Commitment occurs when a user is forced to make a decision before the necessary information is available.

Viscosity is resistance to change, or a measure of how much work is needed to achieve a change.

Abstraction Gradient. An abstraction is the grouping of objects into one object which can make a system easier to understand.

FIGURE 59.4. Example cognitive dimesions.

tunistic where high and low level decisions are mingled; commitment can be weak or strong, and development in one area may be postponed because of problems foreseen in other areas. The vocabulary of dimensions may act as a discovery resource, sensitizing analysts to generic interaction difficulties. However, once discovered, there appears to be little further support for confirmation or elimination

There is little guidance on applying the dimensions. A notation called "ERMIA" (Green, 1991) was proposed as a method of exploring dimensions. Interestingly, this notation seems to have been developed (Green & Benyon, 1995, 1996) into a stand-alone UIM without dimensions.

Pluralistic Walkthrough

The pluralistic walkthrough (Bias, 1994) UIM originated at IBM. An administrator leads a team of three diverse participants (a representative user, a product developer, and a usability specialist) who pretend to be system users. Each is given a set of hard copies of screens and task descriptions. For each screen, the participant is asked to write down in as much detail the next action (or set of actions) the user would take to achieve the task, and any comments they have about the screen. Once each participant has written down the next action, the screen is discussed. When the representative users have finished talking (for example, about problems), the usability specialists and product developers talk (for example, to explain the rationale of certain features). No assessment of this UIM has been published; however, there are clear potential benefits in preparation (via hard copies of screens and task descriptions), and the combination of representative users, product developers, and usability specialists is bound to extend the range of available discovery and analysis resources.

ASSESSING AND SCOPING METHODS

Six methods have been described and reviewed for their support for analyst preparation, problem discovery, and problem analysis. This review indicated where each method provides, or omits, support for three key logical phases in usability inspection: preparation, discovery, and analysis. However, such support should be reflected in the practical effectiveness of a UIM. This section presents and applies the main approaches to assess UIM effectiveness

There are two broad approaches—one qualitative and the other quantitative—to assessing and scoping UIMs. Qualitative methods *code* predictions as a route to grounding explanatory models of analyst performance. Quantitative methods focus on the relative performance of one method to another, generally expressed as a percentage or score (out of one, one being perfect) relating to the proportion of correctly predicted problems. Three such quantitative measures are thoroughness, validity, and efficiency, which are now described before addressing further measures.

Thoroughness, Validity, and Effectiveness

Ideally, UIMs should predict all important user difficulties; for example, we want them to be *thorough*, and thus not miss severe usability problems. UIMs should also ideally predict only genuine problems; for example, we want them to be *valid* and not to generate floods of false positives. Together, we do not want thoroughness at the expense of poor validity; for example, we do not want UIMs to discover almost all the important user difficulties by setting such low thresholds of probability in analysis that most predictions are false positives. We want effective UIMs, we want efficient UIMs, in regards to usage and learning costs, and we want downstream utility (John & Marks, 1997), that is, beneficial influence on design changes. However, these currently have received far less attention than the quality of the problem sets that a UIM produces. The maturity of problem set based measures is reflected in Sears' (1997) formulae for thoroughness, validity, and effectiveness:

$$\text{Effectiveness} = \text{Thoroughness} \times \text{Validity}$$

where

$$\text{Thoroughness} = \frac{\text{number of real usability problems found by UIM}}{\text{number of real problems that exist}}$$

and

$$\text{Validity} = \frac{\text{number of real usability problems found by UIM}}{\text{number of problems predicted by UIM}}$$

Each score is between zero and one, and thus effectiveness will be reduced even for high thoroughness by low validity. This penalty can be increased or decreased by weightings to reflect different evaluation contexts (Hartson, Andre, & Williges, 2001). However, two of the values in the other equations present severe research challenges. Firstly, the denominator for thoroughness (number of real problems that exist) relies on the implicit closures of "all of the usability problems for a design." It should be clear that "all" here could never be determined with confidence: there can always be undiscovered problems. Thus, all thoroughness measures are always *upper bounds*, such as a maximum that will reduce as newly discovered real problems increase the denominator. Note that problems can be lost after they have been discovered, either through poor merging of problem sets from several UIMs or through missing problems in user test data (Cockton & Lavery, 1999). Most analysts miss most problems in video data (Jacobsen, Hertzum, & John, 1998).

Similarly, the numerator for validity (number of real usability problems found by UIM) is a subset of the thoroughness denominator, that is, the overlap between predicted and real problems. The closure here is a subset, since we need not discover all problems, only those that were predicted by the UIM. All validity measures are always *lower bounds*, such as a minimum that increases if newly discovered real problems match predictions and thus increase the numerator.

True closures only exist in algebra, where sets may be closed under operations. Closures are thus wholly synthetic. In the real world, we use the word *all* for enumerations, such as "prohibited items in carry-on baggage." Closure here is the result of human agreement, and even then we find lists of very broad categories rather than specific instances (e.g., ammunition; automatic weapons; axes and hatchets, unless part of aircraft equipment;

billiard cues; billy clubs; and blackjacks). This raises two further challenges for effectiveness metrics for UIMs.

First, we expect lists of usability problems to be more specific than menu problems, layout problems, dragging problems, and difficulties with terminology. We thus cannot even impose a high-level closure on what can be listed as a usability problem. Second, specificity of description determines problem count according to Cockton and Lavery (1999). As we become more specific about usability problems, the count rises. For example, "misleading status bar messages" is one problem, but "incomplete prompts" and "technical vocabulary in message" are two, and more would result if we distinguished between prompt problems such as "no instruction to close free form with double click" or "no instruction to use control-click to delete vertices" (for drawing objects). The limit of problem counts is the full list of all problem instances for all users in all contexts (generalizations of instances are sometimes called "problem tokens;" e.g., John & Mashyna, 1997). Thus, the choice of abstraction level for problem types or tokens is critical for determining the counts of real usability problems.

Thus, problem counts are absolutely determined by basic problem granularity. In theory, effectiveness measures could be made at several levels of abstraction, based on difficulty generalization in the SUPEX method (Cockton & Lavery, 1999), but no one has attempted this yet. However, as long as predicted and real problem sets are expressed at the same level of abstraction, thoroughness and validity measures should not be distorted by categorization.

The issue of comprehensiveness for the set of problem instances is independent of the problem categories. For thoroughness, asymptotic testing has been proposed as a means to ensure the highest possible denominator (Hartson, Andre, & Williges, 2001). However, this is generally understood as continuing to test additional users using the same test protocol until no new problems emerge. Changing the test protocol, such as between different sets of fixed tasks, between free and fixed tasks, or between individual think aloud and peer tutoring, would also reveal new problems. Also, adding analysts and/or applying a structured method such as SUPEX could extract more problems from video data. As with most attempts at closure, methods to find all problems simply expose a further unachievable closure, that is, in *all of the ways* to find all the usability problems.

Maximizing the numerator for validity is more tractable. When real problems are found by fixed task testing, a *falsification* methodology can expose users repeatedly to predicted problems (Woolrych, Cockton, & Hindmarch, 2004). Missed problems here will have two types of cause: errors in test task planning (which can be fixed) and an inability to put the system in a state where the problem would appear (which can be very difficult to achieve; Lavery & Cockton, 1996).

The components of effectiveness are thus unavoidably subject to upper bounds for thoroughness and lower bounds for validity. Where real problems are established by empirical methods, such as user testing, asymptotic and falsification testing can be used to respectively drop the maximum for thoroughness and raise the minimum for validity. Where real problems are merely the union of the problem sets from a range of compared methods, new sources of error arise. While thoroughness can still be calculated as the proportion of all predicted problems found by a single UIM, validity requires reliable methods for identifying false positives. Falsification testing, based on empirical methods, is the only way to identify and manage sources of error here.

Other Quantitative Measures for UIMs

Effectiveness measures depend on creating, counting, and comparing problem sets. Other quantitative measures are based on coding within a problem set (as opposed to coding through comparison, as with false positives, which are in the predicted but not the real problem set). Thus, Cockton and Woolrych (2001) coded heuristic applications for nonbogus predictions as being appropriate or inappropriate, based on explicit criteria from the HE training manual (Lavery, Cockton, & Atkinson, 1996a) provided for analysts. This provided conformance questions that stated, "What the system should do, or users should be able to do, to satisfy the heuristic." (p. 4). Such questions are answered with conformance evidence, such as the "design features or lack of design features that indicate partial satisfaction or breaches of the heuristic." (p. 4). For many heuristic applications, these criteria were clearly ignored. Only 39% of the heuristics applied to nonbogus predictions were appropriate (and only 31% for successful predictions). Such measures of *appropriateness* are important in identifying the extent to which problem discovery and analysis are due to a UIM or to the analysts' own knowledge resources and inspection heuristics. Heuristics tended to be best applied to predictions that turned out to be of low frequency and/or severity. These were likely to be predictions that seemed less probable than ones that turned out to be of high frequency and/or severity, which presumably were so obvious that the 13 heuristics (Nielsen, 1994a, plus three specific to visualization; Lavery, Cockton, & Atkinson, 1996c) were not properly reviewed to strengthen analysis. Furthermore, given the known limited coverage of Nielsen (1994a) heuristics, there may have been *no appropriate heuristic* for some predictions.

Other studies have classified usability problems by severity (e.g., Jeffries, Miller, Wharton, & Uyeda, 1991) and by type. Types may be based on a range of constructs, for example, by relation to stages in Norman's (1986) theory of action. Using this coding scheme, a study (Cuomo & Bowen, 1992, 1994) found that an early version of CW tended to focus on problems related to the formation of actions, and was poor at finding problems concerned with evaluating the display. A similar coding approach was used for an assessment of a later version of CW (Sears & Hess, 1999). This study examined the impact of short and (very) detailed task descriptions on analyst performance. It only reports thoroughness measures, but demonstrates the role of discovery resources (here task descriptions) in shaping analyst performance. The impact of each resource was assessed by coding problems by the CW question relevant to the prediction. Short task descriptions resulted in relatively more predictions related to finding actions (cognitive walkthrough, question 2, Fig. 59.3). Detailed descriptions resulted in relatively more feedback related predictions (question 4, Fig. 59.3). Sears and Hess (1999) based their explanations on discovery task. Detailed (to the level

of naming and locating controls) task descriptions led to oversights in finding action specification problems, but left analysts with the energy to look for feedback problems.

Coding schemes have the benefit of being able to expose why some problems are missed or why some fail to be eliminated during analysis. Problem *discoverability* was coded in an assessment of HE (Cockton & Woolrych, 2001). The easiest problems to discover are *perceivable*: these can be seen at a glance. Next come *actionable* problems: these can be discovered after a few simple actions (e.g., a mouse click). Hardest to find are *constructable problems*, where only several interaction steps involving multiple application objects may reveal a problem. This allowed investigation of the relationship between predictions and discoverability. Eighty percent of missed problems were constructable, as opposed to seven percent of successful predictions. This indicates that task knowledge is not well supported as a discovery resource by HE, confirming that poor integration of context of use information into UIMs is a major source of inspection risk.

UIM assessment is still at an early stage. As researchers develop novel coding schemes to analyze predicted and actual problem sets, new insights become possible about the role and use of discovery and analysis resources in usability inspection. These insights tend to be more valuable than simple thoroughness or validity measures (Woolrych & Cockton 2000), since they can identify which sort of problems get missed (and thus reduce thoroughness) and which predictions fail to be correctly eliminated (and thus reduce validity). As the nature of missed problems (genuine misses and false negatives) and false positives become better understood, researchers and practitioners can explore ways to improve UIMs, either through specific changes to specific methods (fix the method), improved analyst education and training (fix the analyst) or generic improvements that are applicable to a range of methods (fix inspection).

Qualitative Analysis of UIM Usage

Coding of UIM predictions can be used as a basis for quantitative analysis of attributes such as appropriateness and discoverability, based on counts of problems with different codings. Alternatively, coding can be used to develop models of UIM usage, without recourse to counts and statistical inference. Given that most UIM usage variations are due to the analyst and not to UIMs, such qualitative research provides a better basis for UIM improvement, not only via changes to specific UIMs, but through more generic improvements to the inspection process or through analyst training and education. Qualitative analysis was used to refine the DARe model developed by two of this chapter's authors.

Methodological Issues in UIM Assessment

Inescapable upper bounds for thoroughness and lower bounds for validity are two methodological problems for assessing and comparing UIMs. Further methodological problems arise when UIMs are assessed or compared without recourse to user testing based on falsification and asymptotic approaches, since there is no well-grounded way to identify false positives in such studies. Great care should be taken when relying on the results of any UIM assessment or comparison, but no validity measures with any acceptable degree of credibility are possible if empirical methods are not used to construct the actual problem set (the real problems in Sears', 1997, equations). The common approach of comparing predictions from a single UIM to the combined predictions of a set of UIMs is so fraught with methodological challenges that little of any credibility can be concluded from it. This and further methodological flaws formed the basis for an extensive critique of evaluation method assessment (Gray & Salzman, 1998).

Gray and Salzman (1998) reviewed the validity of five major studies in the literature (including, Jeffries et al., 1991; Desurvire, Kondziela & Atwood, 1992; Nielsen, 1992). They based their review on Cook and Campbell's (1979) four main types of validity: statistical conclusion validity, internal validity, construct validity, and external validity. They added a fifth type: conclusion validity. Gray and Salzman found validity problems with each study reviewed. The forms of (in)validity and examples will now be described.

Statistical Conclusion Validity

Was the change to the dependent variable caused by the manipulation of the independent variable? Statistical tests are used to determine if the change did not occur by chance, for example, by wildcard participants, who always do significantly better or worse than the average despite the conditions. Recruiting more participants reduces the wildcard effect. For example, Jeffries et al. (1991) compared CW, HE (actually expert review), guidelines, and user testing and found HE superior. The study used few evaluators, and thus, it is not clear if the effects of the study (for example, the superiority of expert review) occurred by chance.

Internal Validity

Was the effect caused by the variable under manipulation or by a third unknown confounding variable? There is no simple mechanical test for internal validity, but study design must consider instruments, selection of participants and settings, and the use of controls where appropriate. One study where expert review was shown to be superior potentially suffers from internal validity because the experts were given a two-week period at their own paces to complete the evaluation, whereas evaluators in the other conditions had far less time. The worst performing methods were used by software engineers with less usability expertise than the HE group.

Construct Validity

Does the study manipulate the conditions it claims to (causal construct validity) and does it measure the changes it claims to measure (effect construct validity)? Causal construct invalidity would occur if participants in a study used a UIM differently to its documented method. Practitioners need to be informed how different ways of applying UIMs affect the number and

types of problems found and which UIMs are most effective for which software domains.

One study (Karat, Campbell, & Fiegel, 1992) attempted to address causal construct validity, addressing "mono-operation" (single analyst) bias by comparing individual with paired inspections. Similarly, they addressed "monomethod" bias by evaluating two different software systems. Unfortunately they did not specify the differences in the software being evaluated, reducing the usefulness of this study for practitioners. Also, the evaluation had two phases: self-guided exploration and prescribed scenarios. Gray and Salzman (1998) claimed that this study's construct validity was threatened by experience gained in self-guided exploration in the scenario phase.

For "effect construct validity," different evaluation methods focus on either *intrinsic attributes* of the interface (causes) or *pay-off measures* of usability (e.g., effects such as performance time or errors). However, "None of the studies we reviewed report systematic ways of relating pay-off problems to intrinsic features, all apparently rely upon some form of expert judgment" (Gray & Salzman, 1998, p.216). Although they "believe that effect construct validity is the single most important issue facing HCI researchers and practitioners," they gave this little coverage.

External Validity

Do the results of the study generalize to a subpopulation or across a subpopulation? An external validity problem would occur if the results were generalized beyond the settings in the experiment.

Conclusion Validity

Gray and Salzman (1998) introduced a fifth type of validity: conclusion validity. A study is conclusion invalid either if the claims were not investigated in the study or if the data presented in the study contradicted their claim. In one study, Nielsen (1992) investigated the effects of evaluator expertise on the number of problems identified using HE and claimed, "usability specialists were much better than those without usability expertise by finding usability problems with heuristic evaluation." Gray and Salzman claimed this study has conclusion invalidity because it is not clear what the effect of HE was on the evaluator's ability to find usability problems or how many problems evaluators would find through expert review. They suggested the data supported the more modest claim that "experts named more problems than nonexperts." Gray and Salzman found that many studies presented conclusions that were not supported by the data. They did not argue against presenting advice based on experience rather than experimental evidence, but argued that the source of such advice should be made explicit.

Improving UIM Assessment and Comparison

Gray and Salzman's (1998) critique meant that we can trust few existing simple *comparisons* of UIMs, and thus a claim that HE as a UIM finds more problems than CW could lack credibility. However, studies are still relevant to the extent that they (strongly) suggest ways in which specific discovery and analysis resources influence outcomes. Still, it would be better if we had assessments that we can trust, so we first consider what is required here.

Two of Gray and Salzman's (1998) challenges are easily met: (statistical) *conclusion validity* is a fundamental scientific concern. Researchers should not make unsupported claims (and referees and editors should spot them). Due caution and candor are all that is required to avoid writing invalid conclusions. Due vigilance is required to prevent their publication. For *statistical conclusion validity*, seeking competent statistical advice *before* carrying out studies is enough.

In contrast, *external validity* is a challenge for most psychological experiments. This follows from the controls that ensure internal and construct validity. However, if these are not properly addressed, then there can be no external validity, since this can result in unrealistic settings that cannot generalize. Internal and construct validity are logically prior to external validity. Safe generalizations cannot be made from unsound experiments.

Causal construct validity can be addressed by ensuring that assessed UIMs are actually applied by analysts and that they result in predictions. At the very least, UIM use should be compared to unstructured expert judgment, but this practice is very rare (one exception being Bastien & Scapin, 1995). Where this is not done, it may still be possible to establish whether the assessed UIM method actually resulted in the prediction. For example, HE cannot be claimed to have supported a prediction if cited heuristics for it are inappropriate as seen in Cockton and Woolrych (2001).

Effect construct validity is a question of good experimental design, ensuring that instruments and measures in a study really do report the true value of theoretical dependent variables. In too many studies, the main dependent variable has been the percentage of accurate predictions. However, as already argued, 100% cannot be known with certainty, and missed problems (genuine misses and false negatives) may be highly significant, as may false positives. Lastly, *internal validity* requires control of five confounding variables, which we now describe.

Analyst Training and Preparation

This confound is a direct consequence of the impact of the analyst preparation phase. Analysts who do not understand a UIM, the system under evaluation or the expected context of its use cannot be expected to generate sensible predictions. Analysts who do not understand a UIM can readily predict many false positives and miss problems, which would misrepresent a UIM's quality. Studies must carefully control and assess analysts' understanding. It should not be possible to wholly attribute inappropriate UIM use and its impact to analyst misunderstanding.

Reliability of Prediction Merging

When assessing UIMs, it is essential to use a range of analysts. Their predictions must be merged to produce a single problem set. This procedure must be reliable and carefully controlled (Lavery, Cockton, & Atkinson, 1997). Errors in forming a single

set of predictions for each and all analysts will corrupt the problem count, as well as distorting the counts of (in)appropriate UIM applications. Structured problem report formats (Lavery & Cockton, 1997b) have been shown to increase control over prediction merging (Cockton & Woolrych, 2001).

Systematic matching rules require structured usability problems. The individual components of a usability problem report must be reported separately. However, earlier problem report formats (Jeffries, 1994; Mack & Montaniz, 1994; John & Packer, 1995) either restricted the types of problems that can be recorded, missed some crucial aspect of a problem, or were unstructured and hence offer no guidance. Overall, problem matching in existing studies has been too liberal and sometimes based on such crude tactics as matching any problems that concern the same design feature (Cockton & Woolrych, 2001).

Ability of User Testing to Expose Actual Problems

Constructing a real problem set is best achieved through user testing, which brings challenges. Test design and participants will affect the number and types of usability problems discovered. By missing usability problems, both UIMs and user testing may misreport product quality, depending upon the metrics and methods employed. When user test results are used to validate UIMs, failures in user test design will result in miscoding (as false positives) problems that a UIM can predict.

Reliability of Known Problem Extraction

Covering a wide range of problems in user testing does not ensure that such problems will reach the final documented problem set. Errors in forming the set of actual problems from user testing will have predicted problems miscoded as false positives. The SUPEX method (Cockton & Lavery, 1999) addresses this extraction problem. Informed interpretation of problem (type) counts requires explicit levels of problem generalization. Similarly, confidence in coding predictions as (un)successful depends on the abstraction level for the actual problem set. SUPEX also addresses these issues of abstraction and generalization. However, SUPEX is very labor intensive and has yet to be used in UIM assessment. Still, problem extraction is a known methodological problem (Jacobsen et al., 1998). Alternatives to SUPEX involve analysts comparing and resolving problem set differences (Vermeeren, Kesteren, & Bekker, 2003)

Reliability of Matching Predicted to Actual Problems and Structured Report Formats

To assess a UIM, predictions must be matched against actual problems. Errors in matching risk miscoding predicted problems as false positives or vice versa. Existing approaches to problem matching have been either ad hoc or specialized. Ad hoc approaches rely on subjective judgment, and hence are unlikely to be replicable, and therefore a true measure of the method's quality may not be reported. Existing specialized approaches (Lewis et al., 1990; Mack & Montaniz, 1994) either are intended for only one UIM (and so do not generalize), and/or ignore some crucial aspects of usability problems. An investigation of the effects of independent problem matching found evidence of matcher bias (Lavery, Cockton, & Atkinson, 1997), and recommended structured problem reports as a way of reducing this. Without such reports, problem counts are likely to be so unreliable that few calculations in UIM assessment can be trusted. Connell and Hammond (1999) extensively supported this view in their assessment of HE, where they show that the impact of multiple analysts on cumulative problem counts depends on problem merging, resulting in a graph that is less optimistic than that in (Nielsen & Landaur, 1993) and that the handling of problem instances versus problem types can inflate problem counts in published studies. Connell and Hammond showed how this would increase the estimates for test participant needs derived from problem sets, specifically the one used in Nielsen (1994b). Replicable and robust assessments of UIMs thus need matching rules applicable to a wide range of methods. Such rules can only be based on structured problem formats.

There is no universal method for usability problem reporting, but our structured problem report format (SPRFs; Fig. 59.5) aids problem reporting and merging (Cockton & Woolrych, 2001).

SPRFs provide multiple points of reference for problem merging, both to form a master set of problem types from multiple analyses and to compare predicted to actual problems, reducing the potential for miscoding predictions as false (or true) positives. Our SPRF requires *each* analyst to provide a brief description for *every* problem, as well as likely difficulties the user may encounter because of the problem and the specific contexts in which problems may occur. The detail and accuracy of true positives, even from final year undergraduate students, is surprising. The laziest false positives however can be even more disappointing in the stark clarity of a SPRF.

Controlling Confounding Variables

Studies by the authors (Cockton & Woolrych, 2001; Cockton, Woolrych, Hall, & Hindmarch, 2003) have made systematic attempts to control the above five confounding variables. While control of all the variables can be further improved, these studies show what is possible for improving study quality. Three forms of training addressed analysts' understanding of HE. Controlling this variable aimed to improve causal construct and internal validity. As noted, use of SPRFs eases merging of analyst predictions and improves effect construct validity (errors in problem set construction result in errors in effect measurement). The ability of user testing to expose actual problems was addressed by systematically deriving task sets for user testing from analyst predictions, again improving effect construct validity through falsification testing (Woolrych, Cockton, &

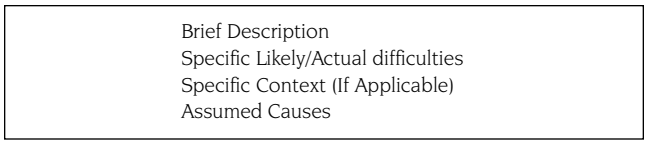


FIGURE 59.5. Sunderland structured problem report format.

Hindmarch, 2004). Reliability of actual problem extraction was addressed by use of general principles from the SUPEX method (Cockton & Lavery, 1999), such as transcription, segmentation, and difficulty isolation and generalization (again, full use of SUPEX remains too labor intensive). Lastly, reliability of matching predicted to actual problems was addressed by directly associating actual with predicted problems. SPRFs ease such associations. Use of a report format for both predicted and actual problems (Lavery & Cockton, 1997b) would further improve effect construct validity.

Outstanding Methodological Challenges

Because of improving internal validity, external validity is improved by supporting focused and accurate conclusions. Further support however is required from extensive coding of accurate predictions, unpredicted problems, and false positives. Coding schemes need to use key usability concepts to provide practitioners with scoping as well as scoring of methods, for example, the problems that are (not) predicted must be analyzed for patterns in their frequency, impact, discovery resources, and other relevant constructs. In addition, predictions that are due to the assessed UIM must be separated from ones that are not. Thus, Sears' (1997) measures of effectiveness are not enough. To achieve causal construct validity, successful predictions must be shown to be due to the UIM being assessed. To support practice, missed problems and false positives must be analyzed across a range of coding dimensions to see why they arise, and how they can be avoided.

UNDERSTANDING INSPECTION METHODS

Software developers need to know the potential benefits of using a particular method (e.g., the proportion of usability problems predicted), the resources needed (e.g., time to evaluate the system, training time for evaluators, evaluators with particular skills), and the risks (e.g., missing certain types of problems or predicting false usability problems). The factors that may affect the quality of the predictions, for example maturation of the prototype, skill level of the evaluator, or choice of different prototyping media, also need to be understood and communicated through method usage heuristics.

All these factors are best explained via a qualitative model of UIM usage. The first two authors have developed such a model, which is now presented as a basis for understanding and assessing UIMs.

The DARe Model of UIM Effectiveness

The DARe model (Discovery and Analysis Resources, initially DR-AR; Woolrych & Cockton, 2002) was derived from an attempt to understand HE's scope and accuracy (Woolrych, 2001), in particular, the source of false positives. In any predictive method, analysts inspect a design for potential usability problems. Where multiple analysts apply a UIM, their problems must be merged into a *master* problem set. This will invariably contain *false positives* (problems predicted by analysts that are not problems for actual users). Now, these can be only incorrectly *analyzed* problems. They could not have been incorrectly *discovered*, as the only discovery error is a genuine miss—analysis is responsible for all other errors.

Not all false positives are predicted by every analyst, leading to a question: Can some analysts correctly eliminate these through a process of analysis as true negatives? This conjecture arises directly from the DARe model, with its logical phases where analysts first discover possible usability problems, then through analysis either confirm them as probable problems or eliminate them as improbable problems. Fig. 59.6 illustrates DARe's basic form as two phases of discovery and analysis. They are logical phases that may overlap, both in time and within analysts' heads.

This simple model can be enhanced by noting that inconsistencies in individual performance (differences in the problems discovered, successful and false predictions) are due to resources external to UIMs. If analysts were wholly reliant on resources supplied by the UIM, differences in performance could not possibly be as diverse as they are. Further support for this argument is demonstrated by Cockton and Woolrych (2001), where analysts successfully predicted a usability problem but applied an inappropriate heuristic to explain it. This indicates the UIM was not responsible for problem discovery, and at the very least, any problem confirmation was based on HE misapplication. A more likely explanation is that analysts are employing resources external to the UIM.

The two logical phases of discovery and analysis were used in the first section of this chapter but with a preceding phase of analyst preparation. The distinction between problem discovery and analysis was made without justification. The logic behind this has now been explained.

The DARe Model and the Scoping of UIMs

The degree to which UIMs alone influence problem discovery and analysis during usability inspection has been assessed in two large studies of HE (Woolrych, 2001; Cockton, Woolrych, Hall, et al., 2003). These studies showed that factors external to

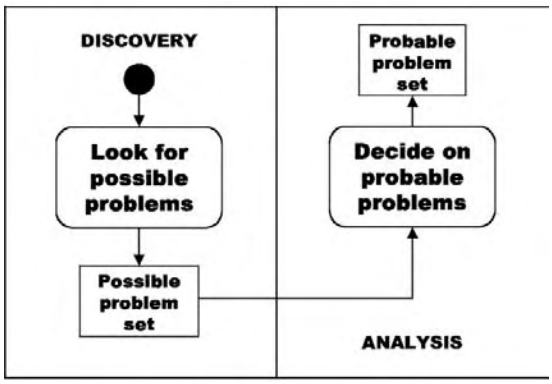


FIGURE 59.6. Simple structure of DARe model.

UIMs influence analysts during inspection. Analysis of predictions reveals *distributed cognitive resources* relevant to analyst performance, especially how misuse of specific distributed cognitive resources leads to false positives. In the first study, Woolrych (2001) found three distinct types of false positives were related to the use of specific resources. The first type, true false positives, generally involved misuse of beliefs about users whereby abilities were underestimated. Analysts predicted possible user difficulties with a particular system feature that did not occur in actual use. Users would often encounter these features, and in the event of any minor difficulties, very quickly recover without problems.

The second type, factually bogus false positives, concerned what analysts believed to be an essential missing feature in the system that would adversely affect the user efficiently performing their tasks. In reality, such features did exist, however, due to limited product knowledge; the analyst was unaware of such features, leading to flawed predictions.

The third type, logically bogus false positives, concerned predictions based on flawed arguments by analysts drawing on design knowledge. Analysts argued that a particular feature would benefit from an alternative design and reverse engineered a rationale that their design was superior. Once again, distributed cognitive resources (here flawed beliefs) external to the UIM were being used.

Development of the DARe model and initial evidence of cognitive resources has motivated further studies and identified more resources (Woolrych, Cockton, & Hindmarch, 2005). So far, seven resource classes have been identified using qualitative coding of problem sets:

- **User** (knowledge of/beliefs about users, especially experience and abilities)
- **Task** (knowledge of what users want to do and how they prefer to accomplish this)
- **Domain** (domain knowledge that is specific to the system being evaluated)
- **Design** (knowledge and experience of interaction design principles, or beliefs in lieu of)
- **Interaction** (knowledge of the how humans actually interact with computers)
- **Technical** (knowledge of platform technologies such as browsers and toolkits)
- **Product** (information about the system and its capabilities)

Fig. 59.7 and Fig. 59.8 extend the simple DARe model by showing how distributed cognitive resources (knowledge, information, and beliefs) are associated with discovery and analysis in usability inspection.

In Fig. 59.7, distributed cognitive resources are associated with system- and user-centered discovery methods. Thus, system-centered methods (e.g., system scanning and searching) predominantly use technical, product, and design resources. Thus, methods such as HE collect design knowledge around heuristics. If HE in any way supports problem discovery, it is through heuristics triggering concerns as, for example, error messages and navigation are inspected. User-centered methods (e.g., goal playing and procedure following) use contextual information about users, tasks, and domains (if not, beliefs must

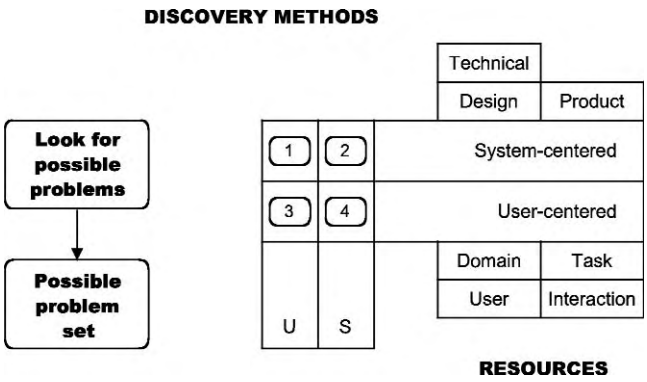


FIGURE 59.7. Cognitive resources and problem discovery in the DARe model (U = Unstructured, S = Structured Discovery Method, 1 = System Scanning, 2 = System Searching, 3 = Goal Playing, 4 = Procedure Following)

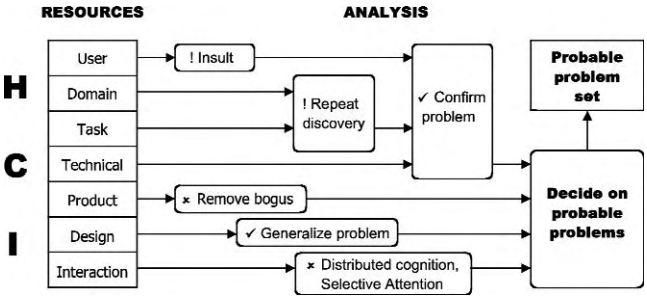


FIGURE 59.8. Cognitive resources and problem analysis in the DARe model.

be substituted) and may integrate knowledge of interaction patterns with system-centered resources. CW relies exclusively on task information to find problems.

Appropriate task, technical, and domain knowledge can aid analysts in valid problem discovery and analysis. Knowledge of tasks allows analysts to discover problems that require sophisticated levels of interaction with the system before they become obvious. Simple system-centered discovery approaches would be unlikely to discover such problems.

Fig. 59.8 shows some example resource usage during problem analysis (confirmation or elimination). Incorrect user beliefs (which are often insults) often lead to a failure to eliminate, and thus to a false positive (hence the ! in Fig. 59.8). Product and interaction knowledge can lead to correct elimination (marked with a cross). Product knowledge gathered during inspection can be used to undo earlier invention of bogus problems. Knowledge of how people interact with computers can imply that apparent design flaws will have no impact, for example, because users will never notice them (selective attention) or because they can learn what they need through interaction (distributed cognition). Technical and design knowledge can be used to correctly confirm a problem (hence the tick). However, reuse of resources in both discovery and analysis introduces risk of confirmation bias. Using the same resources in

discovery and analysis adds nothing, and is thus a source of false positives (Cockton, Woolrych, & Hindmarch, 2004), hence the *repeat discovery* for task and domain knowledge.

The interaction of distributed cognitive resources is the key to successful prediction (Woolrych, Cockton, & Hindmarch, 2005) but still needs to be researched in depth. Work continues to uncover resource interactions, but DARe has already improved understanding of how analysts perform and use UIMs, which is fundamental to any attempt to improve them. Thus, the question arises as to whether a focus on distributed cognitive resources is best exploited via education and training (fix the analyst), via improvements to specific UIMs (fix the method), or via generic improvements to all UIMs (fix inspection). The latter follows from the ubiquitous influence of distributed cognitive resources across all UIMs, suggesting that it may be possible to improve use of resources across all UIMs. However, there are limits to what can be achieved through improvements to usability evaluation. Redesign and downstream utility require separate improvement.

The DARe Model and the Development Lifecycle

Some studies (e.g., Sawyer, Flanders, & Wixon, 1996; John & Marks, 1997; Johnson, 1997) examined the ability (or persuasiveness) of evaluation methods to improve usability, dividing the number of usability problems (or committed to be) fixed by the number of problems reported. However, although the impact of an inspection may be high, there is no guarantee that by addressing its recommendations that the system would be improved—indeed it could be made worse, (e.g., Nielsen, 1993). John and Marks (1997) thus measured design-change (in)effectiveness by whether an implemented recommendation increases, leaves unaffected, or reduces usability problems. They found that only 11% of predicted problems led to an effective change. This downstream utility of UIMs is now becoming a major concern, with passionate claims by Wixon (2003) that the aim of evaluation is to not to find problems but to fix them, a position akin to expecting a medical thermometer to prescribe antibiotics (presumably via an automatic intervening diagnosis). As already argued, UIMs must support effective redesign, through initial extensive causal analysis, but there are many practical reasons why evaluators alone should not be propose design changes (Cockton, 2005), notwithstanding the logical error of expecting a measuring device to repair the object to which it is applied.

The potential value of the DARe model lies in its identification of distributed cognitive resources, which provide extensive effective support for causal analysis and identification of redesign options. However, the main motivation for evaluation in interactive systems development is to drive iterative design. Iteration was original conceived as an attribute of software development processes. However, it is better conceived as a phase that bridges from evaluation to redesign (Cockton, 2005). Iteration takes the results of evaluation, that is, an identification of relevant usability problems along with a partial to near complete understanding of them, and then works towards a redesign to overcome these problems on the basis of a causal analysis. Getting from evaluation to the next iteration of a design is not a simple step, nor should it be. It is illogical to advocate extensive informed grounding of interaction design in contextual research and design rationales during the initial design of an interactive system, and then to expect evaluation methods to generate design change recommendations without again grounding these in business goals, usage contexts, design principles, and design constraints. The only difference between redesign and initial design is that a design has been subjected to evaluation. With UIMs, this produces only one further design input, a list of predicted and partially explained user difficulties. These must be assessed in conjunction with all other design inputs by the whole development team.

Thus, the issue for UIMs is how well they support iteration. It is impossible to attribute downstream utility to the UIM alone. A whole range of factors associated with the development team, process and context will combine to determine which problems get addressed and which ones do not. For example, formal experiments may be required to complete causal analysis, at which point the influence of the UIM will be masked by the impact of the experiments, which could explore alternative layouts or visual designs in a specific interaction context.

It is important to have a balanced view of what UIMs can achieve within the development process, something that applies to all evaluation methods. Attempts to improve usability inspection should be restricted to the proper reporting of predicted problems, with maximum preparation for iteration through causal analysis and identification of redesign options. With this scope in mind, we now consider how UIMs and their usages can be improved.

IMPROVING INSPECTION METHODS

Any improvement in UIMs can only be achieved when what goes wrong in inspection is well understood, which requires continuing advances in research methods to ensure well grounded understandings. Understanding UIMs, and how analysts apply them, is the starting point from where improvements to methods can be made. During inspection, analysts can fail to discover, reject real problems, and keep improbable problems, adversely affecting both thoroughness and validity. Where multiple analysts are involved in inspection, errors in forming the master problem set can also adversely affect a UIM's thoroughness and validity.

Once analysts discover a candidate problem during inspection, a process of analysis takes place where the problem is either kept as a likely problem or rejected as an unlikely problem. Therefore, Woolrych, Cockton, & Hindmarch (2004) determined five possible prediction outcomes for UIMs, of which only the first two are desirable:

1. True positive—discovery of a real problem that is correctly confirmed through analysis, which will cause real difficulties for users (the objective of inspection being to find and report real problems)
2. True negative—a possible problem that is correctly eliminated through analysis (failure to eliminate this would result in a false positive)
3. False positive—a discovered problem that is incorrectly confirmed through analysis (which may result in flawed design iterations)

4. False negative—a discovered problem that is incorrectly eliminated through analysis (undesirable, as without extended report formats, it will be appear as a missed problem)
5. Missed Problems—real problems that exist that analysts fail to even discover

Method-Independent Improvements

Problem discovery is central to a UIM's thoroughness; therefore, appropriate discovery resources and methods are essential. In problem discovery, analysts apply either a system- or user-centered approach (Fig. 59.7). Similarly, each approach can be structured or unstructured. Structured user-centered approaches (task procedure based or goal driven, Fig. 59.7) appeared to be more valid (Cockton, Woolrych, Hall, et al., 2003). Conversely, unstructured, system-centered approaches (simply scanning the interface for potential usability problems, Fig. 59.7) resulted in more false positives. Applying a UIM using a structured user-centered approach resulted in analysts keeping more real problems and fewer false positives. Such improvements may be due to a search strategy that focuses analysts on users and tasks within a usage context; for example, it uses effective discovery resources.

Other issues independent of UIMs also affect their overall performances. Issues associated with multiple analysts, such as problem-merging and problem-reporting methods, also require attention if they are not to negatively impact inspection outcomes. Errors in merging predictions into a single set of predictions for each analyst can affect the problem count.

Multiple analysts are recommended to improve inspection method thoroughness (Nielsen & Landauer, 1993). The DARe model explains why multiple analysts find more problems when the method and environment remain the same. Quite simply, as more analysts are added, more problem discovery resources are added to the inspection, hence more problems are found. However, the DARe model also leads us to considering the impact of multiple analysts on analysis resources. If analysts are inclined to not reject problems at all, or if they are unduly confident in confirming problems, then multiple analysts will not only be collectively more thorough, and thus find more problems, but they will also be collectively less valid, and thus predict more false positives. This conjecture (Woolrych & Cockton, 2002), which is based on an early version of DARe (the DR-AR model), turns out to be true when prediction sets for multiple analysts are examined. This dilution of analysis resources could be counteracted by improved analyst training in UIM usage.

SPRFs (Fig. 59.5) were extended to support our research in UIM assessment, but actually resulted in unexpected improved performance in a study by Cockton, Woolrych, Hall, et al. (2003). To accurately code all analyst predictions, extended SPRFs (ESPRFs) were developed to elicit problem discovery strategies, to distinguish genuine misses from false negatives (a possibility that only arose because of the DARe model), and to distinguish between distributed cognitive resource usage in discovery and analysis (again, a concern that was a direct consequence of the DARe model).

ESPRFs require analysts to indicate the strategy they used to discover individual problems (a combination of unstructured/structured and system/user centered). In addition, all discovered problems (rejected or kept) were reported with relevant justification. This allowed for not only the accurate coding of genuine misses, and true and false negatives, but also identification of distributed cognitive resources influencing such decisions. Surprisingly, ESPRFs resulted in a significant decrease in false positives. Improved validity can be reasonably attributed to the impact of ESPRFs, which encouraged analysts to reflect on discovery methods and to eliminate improbable problems.

Thus, method-independent improvements to UIMs are possible. As well as fixing the method or fixing the analyst, we can fix inspection in general. Surprisingly, tacit training and constraint via ESPRFs proved to be more effective than explicit training through structured tutorials and self-assessment. False positive reduction was mostly due to elimination of bogus problems (both factual and logical), since these are hard to report in ESPRFs. In addition, ESPRFs led analysts to consider problem elimination, which may reduce the negative impact on analysis resources of multiple evaluators. Lastly, the appropriateness of heuristic application also doubled to 61%/65% (mean/median scores for comparison of our 2001 and 2003 studies). (Cockton & Woolrych, 2001; Cockton et al., 2003) All these improvements have been replicated in a further study by Woolrych, Cockton, and Hindmarch (2004), indicating that method-independent ESPRFs can improve validity and appropriateness in UIM usage with a slight improvement in thoroughness (which could be reversed following asymptotic testing).

Method Specific Improvements

Attempts to improve inspection through new and modified UIMs have recently become more common, replacing earlier attempts to combine the best of individual UIMs, as in Sears (1997) heuristic walkthrough. Examples include the usability problem inspector (Hartson, Andre, & Williges, 2003), strategies for improving the effectiveness of HE (Law & Hvannberg, 2004), and metaphors of human thinking (MOT; Frøkjær & Hornbaek, 2002).

MOT was proposed as a new UIM, despite similarities to HE. It was motivated by two perceived flaws in existing UIMs. Firstly, Frøkjær and Hornbaek (2002) claimed that existing techniques fail to consider users' thinking. Secondly, they believed that current UIMs are only useful with specific applications and associated interaction styles (e.g., pointer interfaces), and not appropriate for technologies such as mobile devices, presumably due to dependence on design knowledge specific to desktop graphical user interfaces (although only HE has strong dependencies here).

MOT shifts focus from the device to users' mental activity with the aim of being more applicable to different devices and contexts. In addition, MOT creators have used strikingly poetic language to provide resources in the form of metaphors about human thinking, as shown in Table 59.1, along with their implications for user interfaces. In Fig. 59.1, MOT would appear above walkthroughs, extending beyond these to the right due to its focus on users' mental activity.

Some MOTs are based on knowledge of interaction design from the HCI literature (metaphors 1 and 4); others are based on cognitive psychology (metaphors 2 and 5) or (anti-)philosophizing by a Turing Award winner (metaphors 3, 4, and 5). This very interesting collection of knowledge resources extends ones already identified within the DARe model (knowledge of users and interaction) with inspiration from Naur (1999), often based

TABLE 59.1. Metaphors of Human Thinking

Metaphor of Human Thinking	Implication for User Interface
1. Habit formation is like a landscape eroded by water	Support existing habits and, where necessary develop new ones
2. Thinking as a stream of thought	Users' thinking should be supported by reconcilability, stability, and continuity
3. Awareness as a jumping octopus	Support users' associations with effective means of focusing within a stable context
4. Utterances as splashes over water	Support changes and incomplete utterances
5. Knowing as a site of building	Users should not have to rely on complete or accurate knowledge, design for incompleteness

on admiration for the 19th-century psychologist and philosopher William James.

MOTs are applied to a series of task-based walkthroughs and are claimed to outperform HE (Hornbæk & Frøkjær, 2004). Better performance was due to MOT finding more severe problems that were apparently more complex to repair. In contrast, problems found by HE were more cosmetic and less complex (replicating HE's performance in Cockton & Woolrych, 2001). However, the claimed results may not be the result of the new UIM, and indeed MOT's assessment could be based on unreliable coding of predictions as true/false positives and so forth, since there has been no real problem set derived via empirical user-based methods. Instead, assessment of the problems found by both UIMs was based on an approach by Sawyer et al. (1996) and a related approach by Hartson, Andre, and Williges (2001).

In Hornbæk and Frøkjær (2004), the authors recommended that an individual responsible for the development of the web application under inspection, the client, is responsible for rating the severity of the problems found by each technique. The severity ratings for each problem ranged from very critical (most severe) to not a problem (e.g., a false positive?). Therefore, the clients' judgment is responsible for the identification of false positives, and not more objective and auditable falsification testing (Woolrych, Cockton, & Hindmarch, 2004). The study found the client believed only 1% of problems in reported in HE, and 3% of reported problems in MOT, to be false positives, in marked contrast to 65% false positives in another study of HE (Cockton & Woolrych, 2001). However, in the MOT study, a further 49% of predictions in HE, and 33% in MOT, are described as cosmetic (the lowest severity rating applied, other than those considered not a problem). Assessment of the technique and the accuracy in coding of predictions must be questioned when false positive counts, particularly in HE, are so significantly lower than those in other studies are.

What is more interesting than these potential confounds is how apparent success can be explained by the DARe model. MOT combines task-based search (for problem discovery), with analysis resources from the client who is responsible for much problem coding (e.g., severity, false positives). MOT's thoroughness (with apparent improved problem discovery over HE) may not be a result of its new metaphors, but to how these were applied and how problems are analyzed. MOT is applied as a task-based (procedural) inspection method, in contrast to HE but similar to HW. Analysts familiarize themselves with the application; then, in a manner similar to CW, they attempt typical user tasks in two passes using the metaphors to find usability problems. Once a problem is found, the metaphor responsible for the discovery is associated with the problem as part of the reporting method. However, in HE, analysts often associate inappropriate heuristics with predicted problems (Cockton & Woolrych, 2001). Appropriateness of metaphor association with individual problems was not addressed, leading to questions about their overall impact on problem discovery. Was it the metaphors or other distributed cognitive resources, especially task information, that was responsible for problem discovery and analysis? Also, what benefits derive from the analysis resources provided by a client? A DARe-based critique thus exposes powerful resources: task information required for initial analysis, high quality product, and technical resources from client, plus potential contextual information on users, tasks, and application domain. Put alongside resources of known power and accuracy, it is hard to see what the five metaphors of thinking could add to these problem discovery and analysis resources.

As with previous attempts to improve UIMs directly, initial superficial improvements in validity and thoroughness are as likely (if not more likely) to be a result of method-independent improvements to distributed cognitive resources rather than to new resources specific to the method itself. Thus, MOT as a UIM has strong points in its attempted theoretical grounding in human thinking and its use of structured user-centered discovery methods. However, much of its strength may lie in these last two factors rather than in the metaphors themselves. Similarly, even though heuristics can be applied mostly (and perhaps unavoidably) inappropriately, the use of multiple analysts can still result in thoroughness scores of 70%. This indicates the risks of attributing too much of an inspection's outcome to a UIM, which may have a published coverage of under 40% rather than to the analysts. Given this, we can expect almost any collection of metaphors, heuristics, or principles to appear to give a UIM the power to predict user difficulties. Gray and Salzman's (1998) concerns about conclusion validity remain relevant to all UIM assessments.

Most UIMs are placebos. The ones that are not are ineffective placebo blockers. Evaluators and collaborators, rather than specific methods, may be the main determinant of success, notwithstanding the proven positive benefits of method-independent tools such as ESPRFs and user-centered procedural discovery methods. Thus, the question that arises is about how much further UIM performance could be improved by focused analyst training, ideally based on the actual strategies of the most successful experienced usability professionals.

Improving Analysts

Designing a perfect UIM would appear to be an almost impossible task. The DARe model indicates that a perfect UIM would need the capability to discover candidate problems and confirm or eliminate probable or improbable problems reliably. This means the UIM must be supported by extensive knowledge resources on users, tasks, domains, interaction, design, products, and technologies. Despite efforts to improve UIMs, no new UIM can be claimed to be as reliable as user testing. UIMs can predict more problems, but most of these may be false positives. New

UIMs must rely on human abilities and resources for problem analysis. If we cannot fix the method, and there is little left to fix for inspection in general, can we fix the analyst?

An effective one size fits all UIM is highly improbable, if not impossible, as relevant knowledge resources (and their relative importance) change depending on the application context. For example, domain knowledge in a local travel website is very different to the domain knowledge of a website that hires cars or vans or allows the purchase of electrical goods. By comparison, user knowledge will be a predominant resource in assessing user experience for lifestyle and brand intensive websites, as may currently unexplored knowledge resources for UIMs such as culture and semiotics. Clearly, analysts can become sensitive to and skilled in the use of such resources in advance of their formal integrations into UIMs.

FUTURE TRENDS AND BEST PRACTICE

Prospects for UIMs

Major methodological problems in the assessment and comparison of UIMs have retarded progress in UIM development. Early comparisons and assessments and their limited reliabilities were surveyed in the first version of this chapter (Cockton, Lavery, & Woolrych, 2003). Interested readers should consult this to understand why limited progress was made in the first decade of discount methods. Recent methodological improvements in response to Gray and Salzman's (1998) extensive critique have improved the prospects for UIM improvement, not least through the unexpected side effects of new research instruments such as ESPRFs. In addition, studying analyst behavior rather than method performance is producing insights into effective UIM usage that are relevant to most inspection-based methods. The challenge now is in designing new UIMs, or modifying existing ones, to make best use of generic analyst knowledge, generic inspection heuristics, and method specific procedures. The current renaissance in evaluation method research should bring a stream of improvements throughout the second decade of UIMs.

Best Practice for UIMs

UIM users miss problems, due either to a lack of sufficient discovery resources or to problems found and mistakenly eliminated. In addition, without satisfactory elimination resources such as knowledge of interaction, false positives fail to get eliminated, reducing UIM validity. Neither phase of discovery or analysis in inspection provides enough direct support for analysts. In short, to improve effectiveness, UIMs must find more problems (improved thoroughness), and produce fewer false positives (improved validity). While UIMs rely on multiple analysts to improve thoroughness often at the expense of validity, there are method-independent approaches to better improving overall effectiveness of UIMs.

From a practical point of view, where resources allow, combining the results from inspection with user testing can eliminate false positives. Initially designed as a research tool (Woolrych, Cockton, & Hindmarch, 2004), falsification testing should allow for the accurate coding of all predictions by analysts. Falsification testing involves carefully constructed task sets for user testing following inspection. These task sets follow typical user tasks within the context of the application ensuring analyst predictions are also addressed. Put simply, if a predicted problem exists, falsification testing should find it, or otherwise confirm as a false positive, thus preventing wasted redevelopment time on nonexistent problems. This has the additional benefit of letting UIM users calibrate their predictions against user testing and use this feedback to improve their inspection performances through improvements to reusable cognitive resources and heuristics for carrying out inspections (as opposed to the heuristics used in some UIMs).

Overall, the best practice for all UIM usage is to have a thorough general knowledge of HCI (knowledge of design and interaction); to have accurate and extensive information about the product, its technical platform, and its expected context of use (knowledge of users, tasks and application domain); and to combine these high-quality resources with proven approaches, such as structured problem reports (Fig. 59.5), reflective discovery and analysis of possible and probable usability problems, and preference for structured and/or user-centered approaches to problem discovery (Cockton et al., 2003). UIMs must be thoroughly integrated within an iterative development lifecycle and make full use of all information available to initial design studies. They must also feed into multidisciplinary design iteration, which again makes full use of all available project knowledge and does not shrink back to ill-advised reliance on the judgments of specialist evaluators.

Inspection using multiple analysts needs to be carefully managed. Woolrych and Cockton (2002) found that predictions by a single analyst are not necessarily less probable. Tool and meeting support (potentially asynchronous and remote) needs to be provided for merging problems into a master set. A chief merger may be required, or even a chief analyst or two, leaving all the others to focus on problem discovery. Careful planning and skilled leadership is essentially, regardless of how many analysts are involved. Common advance understandings of appropriate abstraction levels for problem reporting are essential, and analysis must maximize the feed through into subsequent design iterations (causes, fixes).

UIMs are the design equivalent of the bumblebee, which the laws of aerodynamics state should not be able to fly. UIMs, with their limited theoretical underpinnings and their poorly structured procedures and integration into the development lifecycle, should not be able to predict, but they do. Thoroughness scores of 70% and over are common, and if validity can be brought up to the same level, then UIMs may once again become the predominant usability evaluation method. If they do, then this will be due to much improved practice by the ever-improving personal skills and knowledge of thousands of usability specialists, rather than to naïve trust in a handful of HCI researchers and usability gurus.

NOTE

The views and opinions contained in this chapter are those of the authors. In the case of the second author, they do not necessarily represent any official views of Microsoft Corporation. We thank Marc Webster of the School of Computing and Technology, University of Sunderland, for providing all figures.

References

Bastien, J. M. C., & Scapin, D. L. (1995). "Evaluating a User Interface with Ergonomic Criteria", *International Journal of Human-Computer Interaction, 7*(2), pp. 105–121.

Bias, R. G. (1994). "The Pluralistic Usability Walkthrough: Coordinated Empathies", In Nielsen, J. & Mack, R. L. (Eds.), *Usability Inspection Methods,* John Wiley and Sons.

Cockton, G. (2005) "A development framework for value-centred design" In Gerrit C. van der Veer & Carolyn Gale (Eds.) *CHI'05 extended abstracts on Human factors in computing systems,* ACM Press, pp. 1292–195.

Cockton G. & Lavery, D. (1999). "A Framework for Usability Problem Extraction", In Sasse, M.A. & Johnson, C. (Eds.), *Proceedings of Interact '99,* IOS Press, pp. 344–352.

Cockton, G., Lavery D., & Woolrych, A. (2003). "Inspection-based evaluations", In Jacko, J. A. & Sears, A. (Eds.), *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications archive*. Lawrence Erlbaum Associates, Inc., pp. 1118–1138.

Cockton, G. & Woolrych, A. (2001). "Understanding Inspection Methods: Lessons from an Assessment of Heuristic Evaluation", In Blandford, A. & Vanderdonckt, J. (Eds). *People and Computers XV: Interaction without Frontiers*, Springer-Verlag, pp. 171–191.

Cockton, G. & Woolrych, A. (2002). "Sale must end: should discount methods be cleared off HCI's shelves?" *Interactions, 9*(5), pp. 13–18.

Cockton, G., Woolrych, A., & Hindmarch, M. (2004). "Reconditioned Merchandise: Extended Structured Report Formats in Usability Inspection" in Dykstra-Erickson, E., & Tscheligi, M. *CHI '04 extended abstracts on Human factors in computing systems*, ACM Press, pp. 1433–1436.

Cockton, G., Woolrych, A., Hall, L. & Hindmarch, M. (2003) *"Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspection Method Assessment,"* in Palanque, P., Johnson, P. & O'Neill, E. (eds.), People and Computers XVII: Designing for Society, Springer-Verlag, 145–162. John Long Award for Best Paper

Connell, I. W. & Hammond, N. V. (1999) "Comparing Usability Evaluation Principles with Heuristics: Problem Instances vs. Problem Types", In Sasse, M. A. & Johnson, C. (Eds.), *IFIP INTERACT '99: Human-Computer Interaction,* IOS Press, pp. 621–629.

Cook, T. D. & Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings*, Rand McNally.

Cuomo, D. L. & Bowen, C. D. (1992). "Stages of User Activity Model as a Basis for User-System Interface Evaluations", *Proceedings of the Human Factors Society 36th Annual Meeting,* Human Factors Society, pp. 1254–1258.

Cuomo, D. L. & Bowen, C. D. (1994). "Understanding usability issues addressed by three user-system interface evaluation techniques", *Interacting with Computers, 6*(1), pp. 86–108.

Desurvire, H. W., Kondziela, J. M., & Atwood, M. E. (1992). "What is Gained and Lost when Using Evaluation Methods Other than Empirical Testing", In Monk, A., Diaper, D., & Harrison, M. D. (Eds.), *Proc. HCI'92 Conference on People and Computers VII,* Cambridge, pp. 89–102.

Frøkjær, E., & Hornbæk, K. (2002), "Metaphors of human thinking in HCI: Habit, stream of thought, awareness, utterance, and knowing". In R. Kuchinsky, L. Johnson, & F. Vetere (eds.), *Proceedings of HF2002/OzCHI 2002* (CD-Rom). Australia: CHISIG.

Gray, W. D. & Salzman, M. (1998). "Damaged Merchandise? A Review of Experiments that Compare Usability Evaluation Methods", *Human-Computer Interaction, 13*(3), pp. 203–261.

Green, T. R. G. (1991). "Describing Information Artifacts with Cognitive Dimensions and Structure Maps", In Diaper, D. & Hammond, N. (Eds.), *Proceedings of the HCI'91 Conference on People and Computers VI,* Cambridge University Press, pp. 297–315.

Green, T. R. G. & Benyon, D. (1995). "Displays as data structures: entity-relationship models of information artifacts", In Nordby, K., Helmersen, P., Gilmore, D., & Arnesen, S. (Eds.), *Proceedings of INTERACT'95: IFIP TC13 Fifth International Conference on Human-Computer Interaction,* Chapman & Hall, pp. 55–60.

Green, T. R. G. & Benyon, D. (1996.) "The skull beneath the skin: entity-relationship models of information artefacts", *Int. J. Human-Computer Studies, 44*(6), 801–828

Green, T. R. G. & Petre, M. (1996). "Usability analysis of visual programming environments: a 'cognitive dimensions' framework", *Journal of Visual Languages and Computing, 7*(2), pp. 131–174.

Hartson, H. R., Andre, T. S., & Williges, R. C. (2001). "Criteria For Evaluating Usability Evaluation Methods", *International Journal of Human-Computer Interaction, 13*(4), pp. 373–410.

Hartson, H. R., Andre, T. S., & Williges, R. C. (2003) "Determining the Effectiveness of the Usability Problem Inspector: A Theory-Based Model and Tool for Finding Usability Problems" Human Factors: *The Journal of the Human Factors and Ergonomics Society, 3,* pp. 455–482.

Hornbæk, K. & Frøkjær, E. (2004). "Usability Inspection by Metaphors of Human Thinking Compared to Heuristic Evaluation", *Int. J. of Human-Computer Interaction, 17*(3), pp. 357–374.

Jacobsen, N. E., Hertzum, M., & John, B. E. (1998). The evaluator effect in usability tests. In C.-M. Karat & A. Lund (Eds.), *Human Factors in Computing Systems CHI'98 Summary* (pp. 255–256). New York: ACM Press.

Jeffries, R. (1994) "Usability Problem Reports: Helping evaluators communicate effectively with developers", In Nielsen, J. & Mack, R. L. (Eds.), *Usability Inspection Methods,* John Wiley and Sons, pp. 273–294.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. M. (1991) "User Interface Evaluation in the Real World: A Comparison of Four Techniques", In Robertson, S. P., Olson, G. M., & Olson, J. S. (Eds.), *Proc. CHI'91 Conf. on Human Factors in Computing Systems*, ACM, pp. 119–124.

John, B. E. & Marks, S. J. (1997). "Tracking the effectiveness of usability evaluation methods", *Behaviour and Information Technology, 16*(4/5), pp. 188–202.

John, B. E. & Mashyna, M. M. (1997). "Evaluating a Multimedia Authoring Tool", *Journal of the American Society of Information Systems, 48*(11), pp. 1004–1022.

John, B. E. & Packer, H. (1995). "Learning and Using the Cognitive Walkthrough Method: A Case Study Approach", In Katz, I., Mack, R., and Marks, L. (Eds.), *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems,* ACM Press, pp. 429–436.

Johnson, H. (1997). "Generating user requirements from discount usability evaluations", In Harris, D. (Ed.), *Engineering Psychology and Cognitive Ergonomics* (*Volume 2*), Ashgate Publishing Company, pp. 339–357.

Karat, C., Campbell, R., & Fiegel, T. (1992). "Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation", In Bauersfeld, P., Bennett, J., and Lynch, G. (Eds.), *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems,* ACM Press, pp. 397–404.

Lavery, D. & Cockton, G. (1996). "Iterative Development of Early Usability Evaluation Methods for Software Visualisations", In Gray, W. D. & Boehm-Davis, D. A. (Eds.), *Proceedings of the 6th Workshop of Empirical Studies of Programmers,* Ablex, pp. 275–276.

Lavery, D. & Cockton, G. (1997a). "Cognitive Walkthrough: Usability Evaluation Materials", *Technical Report TR-1997-20*, Department of Computing Science, University of Glasgow.

Lavery, D. & Cockton, G. (1997b). "Representing Predicted & Actual Usability Problems", In Johnson, H., Johnson, P., & O'Neill, E. (Eds.), *Proc. Int. Workshop on Representations in Interactive Software Development,* Queen Mary & Westfield College, London, pp. 97–108.

Lavery, D., Cockton, G., & Atkinson, M. (1996a). "Heuristic Evaluation: Usability Evaluation Materials", *Technical Report TR-1996-15*, Dept. of Computing Science, University of Glasgow.

Lavery, D., Cockton, G., & Atkinson, M. (1996b). "Heuristic Evaluation for Software Visualisation: Usability Evaluation Materials", *Report TR-1996-16*, Dept. of Computing Science, University of Glasgow.

Lavery, D., Cockton, G., & Atkinson, M. (1996c). "Cognitive Dimensions: Usability Evaluation Materials", *Technical Report TR-1996-17*, Dept. of Computing Science, University of Glasgow.

Lavery, D., Cockton, G., & Atkinson, M. P., (1997)"Comparison of Evaluation Methods Using Structured Usability Problem Reports," in *Behaviour and Information Technology*, *16*(4), 246–266.

Law, E. L.-C. & Hvannberg, E. T. (2004) "Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation" In *Proceedings of the third Nordic conference on Human-computer interaction*, ACM Press, Tampere, Finland, pp. 241–250.

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990) "Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces", In Carrasco, J. & Whiteside, J. (Eds.), *Proc. CHI'90 Conf. on Human Factors in Computing Systems,* ACM Press, pp. 235–242.

Lewis, C. & Wharton, C. (1997). "Cognitive Walkthroughs", In Helander, M., Landauer, T. K., & Prabhu, P. (Eds.), *Handbook of Human-Computer Interaction* (*2nd Ed*), Elsevier, pp. 717–732.

Mack, R. & Montaniz, F. (1994) "Observing, Predicting and Analyzing Usability Problems", In Nielsen, J. & Mack, R. L. (Eds.), *Usability Inspection Methods,* John Wiley & Sons, pp. 295–339.

Manning, H., (2002), "Reflections: Must the sale end?" *Interactions, 9*(6), ACM, p. 56.

Medlock, M., Wixon, D., McGee, M., & Welsh, D. (2005) "The rapid iterative test and evaluation method: Better products in less time," in R. Bias & D. Mayhew (Eds.), *Cost-Justifying Usability: An Update for the Information Age*, Morgan Kaufman, 489–517.

Muller, M. J., McClard, A., Bell, B., Dooley, S., Meiskey, L., Meskill, J. A., Sparks, R., & Tellam, D. (1995). "Validating an Extension to Participatory Heuristic Evaluation: Quality of Work and Quality of Life", In Katz, I., Mack, R., & Marks, L. (Eds.), *Proc. ACM CHI'95 Conference on Human Factors in Computing Systems* (*Conference Companion*), ACM Press, pp. 115–116.

Naur, P. Reviews of Den originale tænker (Danish edition of Peter Naur: Antiphilosophical Dictionary), Berlingske Tidende, Translations from Danish by Peter Naur available at http://www.naur.com/anmeldelser.html, last accessed 21/12/05.

Nielsen, J. (1992). "Finding Usability Problems Through Heuristic Evaluation", In Bauersfeld, P., Bennett, J., & Lynch, G. (Eds.), *Proc. ACM CHI'92 Conf.* ACM, pp. 373-380.

Nielsen, J. (1993). "Iterative User Interface Design", *IEEE Computer*, *26*(11), pp. 32–41.

Nielsen, J. (1994a). "Enhancing the Explanatory Power of Usability Heuristics", In Adelson, B., Dumais, S., & Olson, J. (Eds.), *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems,* ACM Press, pp. 152–158.

Nielsen, J. (1994b). "Heuristic Evaluation", In Nielsen, J. & Mack, R. L. (Eds.), *Usability Inspection Methods,* John Wiley & Sons, pp. 25–62.

Nielsen, J. (1994c). "Estimating the Number of Subjects Needed for a Think Aloud Test", *International Journal of Human-Computer Studies,* (41), pp. 385–397.

Nielsen, J. & Landauer, T. K. (1993). "A Mathematical Model of the Finding of Usability Problems", In Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E., & White, T. (Eds.), *Proc. INTERCHI'93 Conf. on Human Factors in Computing Systems,* ACM Press, pp. 206–213.

Nielsen, J. & Molich, R. (1990). "Heuristic Evaluation of User Interfaces", In Carrasco, J. & Whiteside, J. (Eds.), *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems,* ACM Press, pp. 249–256.

Norman, D. A. (1986). "Cognitive Engineering", In Norman, D. A. & Draper, S. W. (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction,* Lawrence Erlbaum Associates, pp. 31–61.

Polson, P. G. & Lewis, C. H. (1990). "Theory-Based Design for Easily Learned Interfaces", *Human-Computer Interaction, 5*(2–3), pp. 191–220.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). "Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces", *Int J. Man-Machine Studies, 36*(5), pp. 741–773.

Rieman, J., Davies, S., Hair, D. C., Esemplare, M., Polson, P., & Lewis, C. (1991). "An Automated Cognitive Walkthrough", In Robertson, S. P., Olson, G. M., & Olson, J. S. (Eds.), *Proc. ACM CHI'91 Conf. on Human Factors in Computing Systems,* ACM Press, pp. 427–428.

Rosenbaum, S., Rohn, J. A., & Humburg, J. (2000). "A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys", In Little, R. & Nigay, L. (Eds.), *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems,* ACM Press, pp. 337–344.

Rowley, D. E. & Rhoades, D. G. (1992). "The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure", In Bauersfeld, P., Bennett, J., & Lynch, G. (Eds.), *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems,* ACM Press, pp. 389–395.

Salvucci, D. D., Zuber, M., Beregovaia, E., & Markley, D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. In *Human Factors in Computing Systems: CHI 2005 Conference Proceedings* (pp. 581–589). New York: ACM Press.

Sawyer, P., Flanders, A., & Wixon, D. (1996). "Making A Difference—The Impact of Inspections", In Tauber, M. J. (Ed.), *Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems—Conference Companion,* ACM, pp. 376–282.

Scapin, D. L. (1990). "Organizing Human Factors Knowledge for the Evaluation and Design of Interfaces", *International Journal of Human-Computer Interaction, 2*(3), pp. 203–229.

Scapin, D. L. & Bastien, J. M. C. (1997). "Ergonomic Criteria for Evaluating the Ergonomic Quality of Interactive Systems", *Behaviour and Information Technology, 16*(4/5), pp. 220–231.

Sears, A. (1997). "Heuristic Walkthroughs: Finding the Problems Without the Noise", *International Journal of Human-Computer Interaction, 9*(3), pp. 213–234.

Sears, A. & Hess, D. (1999) "Cognitive Walkthroughs: Understanding the effect of task description detail on evaluator performance", *International Journal of Human-Computer Interaction, 11*(3), 185–200.

Spencer, R. (2000). "The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company", In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 353–359.

Venturi, G. (2004) "Integrating the User Centered approach in the design of Command and Control systems," in Hornbæk, K. & Stage, J., (Eds.) *NordiCHI 2004 Workshop on Improving the Interplay between Usability Evaluation and User Interface Design*, available at http://www.cs.aau.dk/~jans/events/NordiCHI2004WS/venturi.pdf, last accessed 21/12/05

Vermeeren, A., Kesteren, I. V. & Bekker, M. (2003). Managing the Evaluator Effect in User Testing. In *Proceedings of the Interact 2003*, September 1–5, Zürich, Switzerland, pp. 647–654

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). "The Cognitive Walkthrough: A Practitioner's Guide", In Nielsen, J. & Mack, R. L. (Eds.), *Usability Inspection Methods,* John Wiley & Sons, pp. 105–140.

Wixon, D. (2003). "Evaluating usability methods: why the current literature fails the practitioner," in *interactions, 10*(4), 28–34.

Woolrych, A (2001). "Assessing the Scope and Accuracy of the Usability Inspection Method Heuristic Evaluation", *MPhil*, University of Sunderland, available at http://osiris.sunderland.ac.uk/~cs0awo/downloadable%20documents.htm, last accessed 21/12/05

Woolrych, A. & Cockton, G. (2000). "Assessing heuristic evaluation: Mind the Quality, not just Percentages" in S. Turner & P. Turner (Eds.) *Proceedings of British HCI Group HCI 2000 Conference: Volume 2*, British Computer Society: London, pp. 35–36.

Woolrych, A. & Cockton, G. (2002). "Testing a Conjecture Based on the DR-AR Model of Usability Inspection Method Effectiveness" In Sharp, H., Chalk, P., LePeuple, J., & Rosbottom, J. (Eds.), People and Computers XVI: Memorable yet Invisible, Vol. 2, Springer Verlag, pp. 30–33.

Woolrych, A., Cockton, G., & Hindmarch, M. (2004). "Falsification Testing for Usability Inspection Method Assessment," in A. Dearden, & L. Watts (Eds.) *Proceedings of HCI 2004, Volume 2,* Research Press International, pp. 137–140.

Woolrych, A., Cockton, G., & Hindmarch, M. (2005). "Knowledge Resources in Usability Inspection," in L. Mackinnon, O. Bertelsen & N. Bryan-Kinns (Eds), *Proceedings of HCI 2005, Volume 2,* pp. 15–20.

·60·

MODEL-BASED EVALUATION

David Kieras
University of Michigan

Introduction **1192**
What is Model-Based Evaluation? 1192
Organization of this Chapter 1192
Why Use Model-Based Evaluation? 1192
Standard usability design process 1192
Practical limitations of user testing 1192
Theoretical limitations of user testing 1193
The model-based approach 1193
What "interface engineering" should be 1193
Three Current Approaches **1194**
Task Network Models 1194
Cognitive Architecture Models 1194
GOMS Models 1194
Theoretical Basis for Choosing a Model-Based Evaluation Technique **1195**
Psychological Constraints are Essential 1195
A Brief History of Constraints in Modern Psychological Theory 1196
Modeling Cognitive Versus Perceptual-Motor Aspects of a Design 1197
The Science Base Must be Visible 1198
The Value of Generativity 1198
The Role of Detail 1199
Cognitive architectures are committed to detail 1199
Task networks can be used before detailed design 1199
Practical Issues in Applying a Model-Based Evaluation Technique **1200**
Creating the Simulated Device 1200
How to simplify the simulated device 1201
Identifying the Task Strategy 1202
A task analysis does not necessarily specify a task strategy 1202
Difficulties in identifying task strategy 1202
A heuristic: Model what users should do 1202
Concerns Over Model Validity 1203
Can you believe the model? 1203
Should you validate the model? 1203
Summary: Assessing model validity 1204
GOMS Models: A Ready-to-Use Approach **1204**
Why GOMS Models Work 1205
Limitations of GOMS Models 1206
Concluding Recommendations **1206**
References **1206**

INTRODUCTION

What is Model-Based Evaluation?

Model-based evaluation is using a *model* of how a human would use a proposed system to obtain predicted usability measures by calculation or simulation. These predictions can replace or supplement empirical measurements obtained by user testing. In addition, the content of the model itself conveys useful information about the relationship between the user's task and the system design.

Organization of this Chapter

This chapter will first argue that model-based evaluation is a valuable supplement to conventional usability evaluation, and then survey the current approaches for performing model-based evaluation. Because of the considerable technical detail involved in applying model-based evaluation techniques, this chapter cannot include "how to" guides on the specific modeling methods, but they are all well documented elsewhere. Instead, this chapter will present several high-level issues in constructing and using models for interface evaluation, and comment on the current approaches in the context of those issues. This will assist the reader in deciding whether to apply a model-based technique, which one to use, what problems to avoid, and what benefits to expect. Somewhat more detail will be presented about one form of model-based evaluation, GOMS models, which is a well-developed, relatively simple and "ready to use" methodology applicable to many interface design problems. A set of concluding recommendations will summarize the practical advice.

Why Use Model-Based Evaluation?

Model-based evaluation can be best viewed as an alternative way to implement an iterative process for developing a usable system. This section will summarize the standard usability process, and contrast it with a process using model-based evaluation.

Standard usability design process. In simplified and idealized form, the standard process for developing a usable system centers on user testing of prototypes that seeks to compare user performance to a specification or identify problems that impair learning or performance. After performing a task analysis and choosing a set of benchmark tasks, an interface design is specified based on intuition and guidelines both for the platform/application style and usability. A prototype of some sort is implemented, and then a sample of representative users attempts to complete the benchmark tasks with the prototype. Usability problems are noted, such as excessive task completion time or errors, being unable to complete a task, or confusion over what to do next. If the problems are serious enough, the prototype is revised, and a new user test conducted. At some point, the process is terminated and the product completed, either because no more serious problems have been detected, or there is not enough time or money for further development. See Dumas (chapter 57, this volume) for a complete presentation.

The standard process is a straightforward, well-documented methodology with a proven record of success (Landauer, 1995). The guidelines for user interface design, together with knowledge possessed by those experienced in interface design and user testing, adds up to a substantial accumulation of wisdom on developing usable systems. There is no doubt that if this process were applied more widely and thoroughly, the result would be a tremendous improvement in software quality. User testing has always been considered the "gold standard" for usability assessment; however, it has some serious limitations—some practical and others theoretical.

Practical limitations of user testing. A major practical problem is that user testing can be too slow and expensive to be compatible with current software development schedules, so for many years, one focus of HCI research has been on ways to tighten the iterative design loop. For example, better prototyping tools allow prototypes to be developed and modified more rapidly. Clever use of paper mockups or other early user input techniques allows important issues to be addressed before making the substantial investment in programming a prototype. So-called inspection evaluation methods seek to replace user testing with other forms of evaluation, such as expert surveys of the design, or techniques such as cognitive walkthroughs (see Cockton, Lavery, & Woolrich, chapter 59, this volume).

If user testing is really the best method for usability assessment, then it is necessary to come to terms with the unavoidable time and cost demands of collecting behavioral data and analyzing it, even in the rather informal manner that normally suffices for user testing. For example, if the system design were substantially altered on an iteration, it would be necessary to retest the design with a new set of test users. While it is hoped that the testing process finds fewer important problems with each iteration, the process does not get any faster with each iteration—the same adequate number of test users must perform the same adequate number of representative tasks, and their performance assessed.

The cost of user testing is especially pronounced in expert-use domains, where the user is somebody such as a physician, a petroleum geologist, or an engineer. Such users are few, and their time is valuable. This may make relying on user testing too costly to adequately refine an interface. A related problem is evaluating software that is intended to serve experienced users especially well. Assessing the quality of the interface requires a very complete prototype that can be used in a realistic way for an extended period of time so that the test users can become experienced. This drives up the cost of each iteration, because the new version of the highly functional prototype must be developed, and the lengthy training process must be repeated. Other design goals can also make user testing problematic. Consider developing a pair of products for which skill is supposed to transfer from one to the other. Assessing such transfer requires prototyping both products fully enough to train users on the first, and then training them on the second, to see if the savings in training time are adequate. Any design change in either of the products might affect the transfer and thus require a repeat test of the two systems. This double dose of development and testing effort is

probably impractical except in critical domains, where the additional problem of testing with expert users will probably appear.

Theoretical limitations of user testing. From the perspective of scientific psychology, the user testing approach takes very little advantage of what is known about human psychology, and thus lacks grounding in psychological theory. Although scientific psychology has been underway since the late 1800s, the only concepts relied on by user testing are a few basic concepts of how to collect behavioral data. Surely more is known about human psychology than this! The fact is that user-testing methodology would work even if there was no systematic scientific knowledge of human psychology at all; as long as the designer's intuition leads in a reasonable direction on each iteration, it suffices merely to revise and retest until no more problems are found. While this is undoubtedly an advantage, it does suggest that user testing may be a relatively inefficient way to develop a good interface.

This lack of grounding in psychological principles is related to the most profound limitation of user testing: It lacks a systematic and explicit representation of the knowledge developed during the design experience; such a representation could allow design knowledge to be accumulated, documented, and systematically reused. After a successful user testing process, there is no representation of how the design "works" psychologically to ensure usability; there is only the final design itself, as described in specifications or in the implementation code. These descriptions normally have no theoretical relationship to the user's task or the psychological characteristics of the user. Any change to the design, or to the user's tasks, might produce a new and different usability situation, but there is no way to tell what aspects of the design are still relevant or valid. The information on why the design is good, or how it works for users, resides only in the intuitions of the designers. While designers often have outstanding intuitions, we know from the history of creations such as the medieval cathedrals that intuitive design is capable of producing magnificent results, but is also routinely guilty of costly overengineering or disastrous failures.

The model-based approach. The goal of model-based evaluation is to get some usability results before implementing a prototype or testing with human subjects. The approach uses a *model* of the human-computer interaction situation to represent the interface design and produce predicted measurements of the usability of the interface. Such models are also termed *engineering models* or *analytic models* for usability. The model is based on a detailed description of the proposed design and a detailed task analysis. It explains how the users will accomplish the tasks by interacting with the proposed interface, and uses psychological theory and parametric data to generate the predicted usability metrics. Once the model is built, the usability predictions can be quickly and easily obtained by calculation or by running a simulation. Moreover, the implications of variations on the design can be quickly explored by making the corresponding changes in the model. Since most variations are relatively small, a circuit around the revise/evaluate iterative design loop is typically quite fast once the initial model-building investment is made. Thus, unlike user testing, iterations generally get faster and easier as the design is refined.

In addition, the model itself summarizes the design, and can be inspected for insight into how the design supports (or fails to support) the user in performing the tasks. Depending on the type of model, its components may be reusable not in just different versions of the system under development, but in other systems as well. Such a reusable model component captures a stable feature of human performance, task structures, or interaction techniques; characterizing them contributes to our scientific understanding of human-computer interaction.

The basic scheme for using model-based evaluation in the overall design process is that iterative design is done first using the model, and then by user testing. In this way, many design decisions can be worked out before investing in prototype construction or user testing. The final user testing process is required for two reasons: First, the available modeling methods only cover certain aspects of usability; at this time, they are limited to predicting the sequence of actions, the time required to execute the task, and certain aspects of the time required to learn how to use the system. Thus, user testing is required to cover the remaining aspects. Second, since the modeling process is necessarily imperfect, user testing is required to ensure that some critical issue has not been overlooked. If the user testing reveals major problems along the lines of a fundamental error in the basic concept of the interface, it will be necessary to go back and reconsider the entire design; again model-based iterations can help address some of the issues quickly. Thus, the purpose of the model-based evaluation is to perform some of the design iterations in a lower cost, higher speed mode before the relatively slow and expensive user testing.

What "interface engineering" should be. Model-based evaluation is not the dominant approach to user interface development; most practitioners and academics seem to favor some combination of user testing and inspection methods. Some have tagged this majority approach as a form of "engineering;" however, even a cursory comparison to established engineering disciplines makes it clear that conventional approaches to user interface design and evaluation have little resemblance to an engineering discipline. In fact, model-based evaluation is a deliberate attempt to develop and apply true engineering methods for user interface design. The following somewhat extended analogy will help clarify the distinction, as well as explain the need for further research in modeling techniques.

If civil engineering were done with iterative empirical testing, bridges would be built by erecting a bridge according to an intuitively appealing design, and then driving heavy trucks over it to see if it cracks or collapses. If it does, it would be rebuilt in a new version (e.g., with thicker columns) and the trial repeated; the iterative process continues with additional guesses until a satisfactory result is obtained. Over time, experienced bridge builders would develop an intuitive feel for good designs and how strong the structural members need to be, and so will often guess right. Time and cost pressures, however, will probably lead to cutting the process short by favoring conservative designs that are likely to work, even though they might be unnecessarily clumsy and costly.

Although early bridge building undoubtedly proceeded in this fashion, modern civil engineers do not build bridges by iterative testing of trial structures. Rather, under the stimulus of

design failures (Petrosky, 1985), they developed a body of scientific theory on the behaviors of structures and forces, and a body of principles and parametric data on the strengths and limitations of bridge-building materials. From this theory and data, they can quickly construct models in the form of equations or computer simulations that allow them to evaluate the quality of a proposed design without having to physically construct a bridge. Thus, an investment in theory development and measurement enables engineers to replace an empirical iterative process with a theoretical iterative process that is much faster and cheaper per iteration. The bridge is not built until the design has been tested and evaluated based on the models, and the new bridge almost always performs correctly. Of course, the modeling process is fallible, so the completed bridge is tested before it is opened to the public, and occasionally the model for a new design is found to be seriously inaccurate and a spectacular and deadly design failure is the result. The claim is not that using engineering models is perfect or infallible, only that it saves time and money, and thus allows designs to be more highly refined. In short, more design iterations results in better designs, and better designs are possible if some of the iterations can be done very cheaply using models.

Moreover, the theory and the model summarize the design and explain why the design does or does not work well. The theoretical analysis identifies the weak and strong points of the design, giving guidance to the designer where intuition can be applied to improve the design; a new analysis can then test whether the design has actually been improved. Engineering analysis does not result in simply static repetition of proven ideas. Rather, it enables more creativity because it is now possible to cheaply and quickly determine whether a new concept will work. Thus, novel and creative concepts for bridge structures have steadily appeared once the engineering models were developed.

Correspondingly, model-based evaluation of user interfaces is simply the rigorous and science-based techniques for how to evaluate user interfaces without user testing; it likewise relies on a body of theory and parametric data to generate predictions of the performance of an engineered artifact, and explain why the artifact behaves as it does. While true interface engineering is nowhere as advanced as bridge engineering, useful techniques have been available for some time, and should be more widely used. As model-based evaluation becomes more developed, it will become possible to rely on true engineering methods to handle most of the routine problems in user interface design, with considerable savings in cost and time, and with reliably higher quality. As has happened in other branches of engineering, the availability of powerful analysis tools means that the designer's energy and creativity can be unleashed to explore fundamentally new applications and design concepts.

THREE CURRENT APPROACHES

Research in HCI and allied fields has resulted in many models of human-computer interaction at many levels of analysis. This chapter restricts attention to approaches that have developed to the point that they have some claim, either practical or scientific, to being suitable for actual application in design problems. This section identifies three current approaches to modeling human performance that are the most relevant to model-based evaluation for system and interface design: (a) task network models, (b) cognitive architecture models, and (c) GOMS models.

Task Network Models

In task network models, task performance is modeled in terms of a PERT-chart-like network of processes. Each process starts when its prerequisite processes have been completed, and has an assumed distribution of completion times. This basic model can be augmented with arbitrary computations to determine the completion time, as well as what its symbolic or numeric inputs and outputs should be. Note that the processes are usually termed "tasks," but they need not be human-performed at all, but can be machine processes instead. In addition, other information, such as workload or resource parameters can be attached to each process. Performance predictions are obtained by running a Monte-Carlo simulation of the model activity, in which the triggering input events are generated either by random variables or by task scenarios. A variety of statistical results, including aggregations of workload or resource usage, can be readily produced. The classic SAINT (Chubb, 1981) and the commercial MicroSaint tool (Laughery, 1989) are prime examples. These systems originated in applied human factors and systems engineering, and are heavily used in system design, especially for military systems.

Cognitive Architecture Models

Cognitive architecture systems are surveyed by Byrne (chapter 5, this volume). These systems consist of a set of hypothetical interacting perceptual, cognitive, and motor components assumed to be present in the human, and whose properties are based on empirical and theoretical results from scientific research in psychology and allied fields. The functioning of the components and their interactions are typically simulated with a computer program, which, in effect, produces a *simulated human* performing in a *simulated task environment* that supplies inputs (stimuli) to the simulated human, and reacts to the outputs (responses) produced by the simulated human. Tasks are modeled primarily by programming the cognitive component according to a task analysis, and then performance predictions are obtained by running the simulation using selected scenarios to generate the input events in the task. Because these systems are serious attempts to represent a theory of human psychological functions, they tend to be rather complex, and are primarily used in basic research projects; there has been very limited experience in using them in actual design settings.

GOMS Models

GOMS models are the original approach to model-based evaluation in the computer user interface field; both the model-based evaluation approach and GOMS models were presented as

methods for user interface design in the seminal Card, Moran, and Newell (1983) presentation of the psychology of human-computer interaction. They based the GOMS concept on the theory of human problem solving and skill acquisition. In brief, GOMS models describe the knowledge of procedures that a user must have in order to operate a system. The acronym and the approach can be summarized as follows: The user can accomplish certain Goals (G) with the system; Operators (O) are the basic actions that can be performed on the system, such as striking a key or finding an icon on the screen; Methods (M) are sequences of Operators that when executed, accomplish a Goal; and Selection Rules (S) describe which Method should be used in which situation to accomplish a Goal, if there is more than one available. Constructing a GOMS model involves writing out the methods for accomplishing the task goals of interest, and then calculating predicted usability metrics from the method representation.

There are different forms of GOMS models, systematized by John and Kieras (1996a, 1996b), that represent the methods at different levels of detail, and whose calculations can range in complexity from simple hand calculations to full-fledged simulations. John and Kieras pointed out that the different forms can be viewed as being based on different simplified cognitive architectures that make the models easy to apply to typical interface design problems and insulate the model-builder from many difficult theoretical issues. More so than any other model-based approach, GOMS models have a long and well-established track record of success in user interface design, although they are not used as widely as their simplicity and record would justify. Although still under development by researchers, GOMS models are emphasized in this chapter because in some forms they are a "ready to use" modeling methodology. A later section will describe their rationale more completely, but the reader is referred to John and Kieras (1996a, 1996b) for a thorough discussion.

THEORETICAL BASIS FOR CHOOSING A MODEL-BASED EVALUATION TECHNIQUE

This section presents several key issues concerning the theoretical foundations of model-based evaluation, concerning the basic sources of information and applicability of the modeling approach. When choosing or evaluating a technique for model-based evaluation, the potential user should consider these issues; the techniques differ widely in how well they handle certain fundamental questions. The next section will focus on the practical problems of applying a modeling technique once it has been chosen. In both sections, the three basic approaches to model-based evaluation are commented on as appropriate. Advice is given to both the user of model-based evaluation and the developer of model-based techniques.

Psychological Constraints are Essential

The concept of model-based evaluation in system design has a long history and many different proposed methods (for early surveys, see Pew, Baron, Feehrer, & Miller, 1977; MacMillan, Beevis, Salas, Strub, Sutton, & Van Breda, 1989; Elkind, Card, Hochberg, & Huey, 1989); however, the necessary scientific basis for genuinely powerful models has been slow to develop. The key requirement for model-based evaluation is that building a model to evaluate a design must be a routine, production, or engineering activity, and not a piece of basic scientific research on how human psychological factors are involved in a particular computer usage situation. This means that the relevant psychological science must not only be developed first, but also must then be systematized and encapsulated in the modeling methodology itself. A modeling methodology must provide *constraints* on the content and form of the model, and these constraints must provide the psychological validity of the model as a predictor of human performance. In other words, if the model builder can do essentially anything in the modeling system then the only way the resulting model can be psychologically valid is if the model builder does all of the work to construct a valid psychological theory of human cognition and behavior in the task, and then ensures that the constructed model accurately reflects this theory.

Of course it takes tremendous time, effort, and training to construct original psychological theory, far more than should be necessary for most interface design situations. Although the decisions in truly novel or critical design situations might require some fundamental psychological research, most interface design situations are rather routine. The problem is to match a computer system to the user's tasks using known interface design concepts and techniques. It should not be necessary to be an expert researcher in human cognition and performance to carry this out.

Thus, the key role of a modeling system is to provide constraints based on the psychological science, so that a model constructed within the system has a useful degree of predictive validity. In essence, simply by using the modeling system according to its rules, the designer must be able to construct a scientifically plausible and usefully accurate model "automatically."

A simple series of examples will help make the point: Computer user interfaces involve typing of arbitrary strings of text on the keyboard and pointing with a mouse. The time required to type on the keyboard and to point with a mouse are fairly well documented. If task execution times are of interest, an acceptable modeling system should include these human performance parameters so that the interface designer does not have to collect them or guess them.

Furthermore, because both hands are involved in typing strings of text, users cannot type at the same time as they move the mouse cursor; these operations must be performed sequentially, taking considerably more time than if they could be done simultaneously. A modeling system should make it impossible to construct a model of an interface that overzealously optimizes execution speed by assuming that the user could type strings and point simultaneously; the sequential constraint should be enforced automatically. A high-quality modeling system would not only enforce this constraint, but also automatically include the time costs of switching between typing and pointing, such as the time to move the hand between the mouse and the keyboard. There are many such constraints on human performance—some of them quite obvious, as in these examples, and some very subtle. A good modeling system will

represent these constraints in such a way that they are automatically taken into account in how the model can be constructed and used. Because of the subtleties involved, computational tools are especially valuable for constructing and using models because they can help enforce the psychological constraints and make it easier for the model-builder to work within them.

A Brief History of Constraints in Modern Psychological Theory

Theoretical constraints are not easy to represent or incorporate; a coherent and rigorous theoretical foundation is required to serve as the substrate for the network of constraints, and suitable foundations were not constructed until fairly recently. Through most of second half of the 20th century, psychological theory was mired in a rather crude form of information-processing theory, in which human activity was divided into information-processing stages such as perception, memory, decision-making, and action. These stages were usually depicted in a flowchart with a box for each stage, with various connections between the boxes, and perhaps with some fairly simple equations that described the time required for each stage or the accuracy of its processing. There was little constraint, however, on the possible data contained in each box, or the operations performed there. A box could be of arbitrary complexity, and no actual explicit mechanism had to be provided for any of them. Such models were little more than a "visual aid" for theories posed in the dominant forms of informal verbal statements or rather abstract mathematical equations. Later, many researchers began to construct computer simulations of these "box models," which provided more flexibility than traditional mathematical models and also contributed more explicitness and rigor than traditional verbal models. But still, the operations performed in each box were generally unstructured and arbitrary.

An early effort at model-based evaluation in this theoretical mode appears in the famous Human Operator Simulator (HOS) system (see Wherry, 1976; Pew et al., 1977; Strieb & Wherry, 1979; Lane, Strieb, Glenn, & Wherry, 1981; Glenn, Zaklad, & Wherry, 1982; Harris, Iavecchia, & Bittner, 1988; and Harris, Iavecchia, & Dick, 1989). HOS contained a set of *micromodels* for low-level perceptual, cognitive, and motor activities, invoked by task-specific programs written in a special-purpose procedural programming language called HOPROC (Human Operator Procedures language). The micromodels included such things as Hick's and Fitts' Law, formulas for visual recognition time, a model of short-term memory retention, and formulas for calculating the time required for various motor actions such as pushing buttons and walking. The effort was ambitious and the results impressive, but in a real sense, HOS was ahead of its time. The problem was that psychological theory was not well enough developed at the time to provide a sound foundation for such a tool; the developers were basically trying to invent a cognitive architecture good enough for practical application before the concept had been developed in the scientific community. Interestingly, the spirit of the HOPROC language lives on in the independently developed notations for some forms of GOMS models. In addition, the scientific base for the micromodels was in fact very sparse at the time, and many of them are currently out of date both empirically and theoretically. HOS appears to have been subsumed into some commercial modeling systems; for example, a task network version of HOS is available from Micro Analysis and Design, Inc. (http://www.maad.com/), and its micromodels are used in their Integrated Performance Modeling Environment (IPME), as well as CHI System's COGNET/IGEN produced by CHI Systems (http://www.chiinc.com/).

The task network models also originated in this box-model mode of psychology theory, and show it in their lack of psychological constraints; their very generality means they contribute little built-in psychological validity. Even if the HOS micromodels are used, the flexibility of the modeling system means that model builders themselves must identify the psychological processes and constraints involved in the task being modeled and program them into the model explicitly.

Led by Anderson (1983) and Newell (1990) researchers in human cognition and performance began to construct models using a *cognitive architecture* (see Byrne, chapter 5, this volume). Cognitive architecture parallels the concept of computer architecture: a cognitive architecture specifies a set of fixed mechanisms, the "hardware," that comprise the human mind. To construct a model for a specific task, the researcher "programs" the architecture by specifying a psychological strategy for doing the task, the "software." (Parameter value settings and other auxiliary information might be involved as well.) The architecture provides the coherent theoretical framework within which processes and constraints can be proposed and given an explicit and rigorous definition. Several proposed cognitive architectures exist in the form of computer simulation packages in which programming the architecture is done in the form of production systems, collections of modular if-then rules, that have proved to be an especially good theoretical model of human procedural knowledge. Developing these architectures, and demonstrating their utility, is a continuing research activity (see Byrne, chapter 5, this volume). Not surprisingly, they all have a long way to go before they accurately incorporate even a subset of the human abilities and limitations that appear in an HCI design context.

The psychological validity of a model constructed with a cognitive architecture depends on the validity of both the architecture and the task-specific programming, so it can be difficult to assign credit or blame for success or failure in modeling an individual task. However, the fixed architecture and its associated parameters, are supposed to be based on fundamental psychological mechanisms that are required to be invariant across all tasks, while the task-specific programming is free to vary with a particular modeled task. To the extent that the architecture is correct, one should be able to model any task simply by programming the architecture using only task-analytic information and supplying a few task-specific parameters. The value of such architectures lies in this clear division between universal and task-specific features of human cognition; the model builder should be free to focus solely on the specific task and system under design, and let the architecture handle the psychology.

Achieving this goal in psychological research is a daunting challenge. What about the practical sphere? In fact, the role of architectural constraints in some of the extant commercial modeling systems is problematic. The task network models basically

have such an abstract representation that there is no straightforward way for architectural assumptions to constrain the modeling system. Once one has opted for representing human activity as a set of arbitrary interconnected task processes, there is no easy way to somehow impose more constrained structure and mechanism on the system. Attempting to do so simply creates more complexity in the modeling problem: The modeler must figure out how to *underuse* the *overgeneral* capabilities of the system in just the right way.

Another commercial system, COGNET/IGEN (see Zachary, Santarelli, Ryder, & Stokes, 2000, for a recent and relatively complete description), is in the form of a cognitive architecture, but is a very complex one that incorporates a multitude of ideas about human cognition and performance, so many that it appears to be rather hard to understand how it works. The essence of a cognitive architecture, however, is the insistence on a small number of fundamental mechanisms that provide a comprehensive and coherent system. For example, several of the scientifically successful cognitive architectures require that all cognitive processing must be expressed in the form of production rules that can include only certain things in their conditions and actions. These rules control all of the other components in the architecture, which in turn have strictly defined and highly limited capabilities. These highly constrained systems have been successful in a wide range of modeling problems, so it is difficult to see why a very complex architecture is a better starting point. Again, to be useful in both scientific and practical prediction, the possible models must be constrained; too many possibilities are not helpful, but harmful.

From the point of view of cognitive architectures and the constraints supplied by the architecture, the modeling approaches described in this chapter, as currently implemented, span the range from little or no architectural content or constraints (the task network systems) to considerable architectural complexity and constraints (the cognitive architecture systems). GOMS models occupy an intermediate position: they assume a simplified, but definitely constraining, cognitive architecture that allows them to be applied easily by interface designers and still produce usefully accurate results. At the same time, however, they are less flexible than the modeling systems at the other extremes.

Modeling Cognitive Versus Perceptual-Motor Aspects of a Design

As pointed out by Byrne (chapter 5, this volume), cognitive architectures have lately begun to incorporate not just proposed cognitive mechanisms, but also proposals for perceptual and motor mechanisms that act as additional sources of constraint on performance. Calling these "cognitive" architectures is something of a misnomer, since perceptual and motor mechanisms are normally distinguished from cognitive ones. However, including perceptual and motor constraints, is actually a critical requirement for modeling user interfaces; this follows from the traditional characterization of human-computer interaction in terms of the interactive cycle (Norman, 1986). The user sees something on the screen if the user is looking in the right place and can sense and recognize it, which involves the perceptual system and associated motor processes such as eye movements. The user decides what to do, an exclusively cognitive activity, and then carries out the decision by performing motor actions that are determined by the physical interaction devices that are present, and may also involve the perceptual system, such as visual guidance for mouse pointing.

Occasionally, the cognitive processes of deciding what to do next can dominate the perceptual and motor activities. For example, one mouse click might bring up a screen containing a single number, such as a stock price, and the user might think about it for many minutes before simply clicking on a "buy" or "sell" button. Much of the time, however, users engage in a stream of routine activities that require only relatively simple cognitive processing, and so the perceptual and motor actions take up most of the time and determine most of the task structure. Two implications follow from this thumbnail analysis:

1. *Modeling purely cognitive tasks is generally impractical.* Trying to model purely cognitive tasks such as human problem-solving, reasoning, or decision-making processes is extremely difficult because they are so open-ended and unconstrained (see also Landauer, 1995). For example, there are a myriad possible ways in which people could decide to buy or sell a stock, and the nature of the task does not set any substantial or observable constraints on how people might make such decisions. Stock decisions are based on everything from gut feel, to transient financial situations, to detailed long-term analysis of market trends and individual corporate strategies. Trying to identify the strategy that a user population will follow in such tasks is not a routine interface design problem, but a scientific research problem, or at least a very difficult task analysis problem. Fortunately, a routine task analysis may produce enough information to allow the designer to *finesse* the problem—that is, sidestep it or avoid having to confront it. For example, if one could determine what information the stock trader needs to make the decisions, and then make that information available in an effective and usable manner, the result will be a highly useful and usable system without having to understand exactly how users make their decisions.
2. *Modeling perceptual-motor activities is critical.* A good modeling approach at a minimum must explicitly represent the perceptual and motor operations involved in a task. For most systems, the perceptual and motor activities involved in interacting with a computer (a) take relatively well-defined amounts of time, (b) are heavily determined by the system design, and (c) frequently dominate the user's activity; leaving them out of the picture means that the resulting model is likely to be seriously inaccurate. For example, if two interface designs differ in how many visual searches or mouse points they logically require to complete a task, the one requiring fewer is almost certainly going to be faster to execute, and will probably have a simpler task structure as well, meaning it will probably be easier to learn and less error-prone. This means that any modeling approach that represents the basic timing and the structure of perceptual and motor activity entailed by an interface is likely to provide a good approximation to the basic usability characteristics of the interface.

This conclusion is both good news and bad news. The good news is that since perceptual-motor activities are relatively easy to model, it can be easy to get fairly reliable and robust model-based evaluation information in many cases. One reason why GOMS models work so well is that they allow the modeler to easily represent perceptual-motor activity fairly completely, with a minimum of complications to represent the cognitive activity. The bad news is that different modeling approaches that include perceptual-motor operations are likely to produce similar results in many situations, making it difficult to tell which approach is the most accurate.

This does not mean continuing the effort to develop modeling systems is futile; rather, the point is that trying to verify or compare models in complex tasks is quite difficult due to practical difficulties in both applied and basic research. Despite the considerable effort and expense to collect it, data on actual real-world task performance is often lacking in detail, coverage, and adequate sample sizes. Even in the laboratory, collecting highly precise, detailed, and complete data about task performance is quite difficult, and researchers are typically trapped into using tasks that are artificial, performed by nonexpert subjects, or trivial relative to actual tasks. There is no easy or affordable resolution to this dilemma, so the practitioner who seeks to use models must be cautious about claims made by rival modeling approaches, and look first at how they handle perceptual-motor activities. The theorist seeking to improve modeling approaches must be constantly iterating and integrating over both laboratory and actual applications of modeling methods.

The Science Base Must be Visible

Even though the modeling methodology encapsulates the constraints provided by psychological theory, it is critical that the psychological assumptions be accessible, justified, and intelligible. An architecture is the best way to do this, because the psychological assumptions are either hardwired into the modeling system architecture, or are explicitly stated in the task-specific programming supplied by the modeler. The basis for the task-specific programming is the task analysis obtained during the overall design process, and the basis for the architecture is a documented synthesis of the scientific literature.

The importance of the documented synthesis of the scientific literature cannot be overstated. The science of human cognition and performance that is relevant to system design is not at all "finished;" important new results are constantly appearing, and many long-documented phenomena are incompletely understood. Thus, any modeling system will have to be updated repeatedly as these theoretical and empirical issues are thrashed out, and it will have to be kept clear which results it incorporates and which it does not.

The commercial modeling tools have seriously lagged behind the scientific literature. While some conservatism would be desirable to damp out some of the volatility in scientific work, the problem is not just conservatism, but rather obsolescence, as in the case of the micromodels inherited from HOS. Perhaps these systems would still be adequate for practical work but, unfortunately, it is very difficult to get a scientific perspective on their adequacy because they have been neither described nor tested in forums or under ground rules similar to those used for mainstream scientific work in human cognition and performance. Thus, they have not been subject to the full presentation, strict review, criticism, and evolution that are characteristic of the cognitive architecture and GOMS model work. The practitioner should therefore greet the claims of commercial modeling system with healthy skepticism, and developers of modeling systems should participate more completely in the open scientific process.

The Value of Generativity

It is useful if a modeling method is *generative*, meaning that a single model can *generate* predicted human behavior for a whole class of scenarios, where a scenario is defined solely in terms of the sequence of input events or the specifications for a task situation, neither of which specifies the behavior the user is expected to produce. Many familiar modeling methods, including the Keystroke-Level type of GOMS model, are nongenerative, in that they start with a specific scenario in which the model builder has specified, usually manually, what the user's actions are supposed to be for the specified inputs. A nongenerative model predicts metrics defined only over this particular input-output sequence. To see what the results would be for a different scenario, a whole new model must be constructed (though parts might be duplicated). Since nongenerative modeling methods are typically labor intensive, involving a manual assignment of user actions to each input-output event, they tend to sharply limit how many scenarios are considered, which can be very risky in complex or critical design problems.

An example of a sophisticated nongenerative modeling method is the CPM-GOMS models developed by Gray, John, and Atwood (1993) to model telephone operator tasks. These models decomposed each task scenario into a set of operations performed by perceptual, cognitive, and motor processors likes those proposed in the Card, Moran, and Newell (1983) Model Human Processor. The sequential dependencies and time durations of these operations were represented with a PERT chart, which then specified the total task time, and whose critical path revealed the processing bottlenecks in task performance. Such models are nongenerative in that a different scenario with a different pattern of events requires a different PERT chart to represent the different set of process dependencies. Since there is a chart for each scenario, predicting the time for a new scenario, or different interface design, requires creating a new chart to fit the new sequence of events. However, a new chart can often be assembled from templates or portions of previous charts, saving considerable effort (see John & Kieras, 1996a, 1996b for more detail). In addition, computational tools for modeling based on directly representing the sequential constraints implied by a cognitive architecture are under development and may substantially simplify the construction of such models (e.g., John, Vera, Matessa, Freed, & Remington, 2002; Vera, Howes, McCurdy, & Lewis, 2004).

If a model is generative, a single model can produce predicted usability results for any relevant scenario, just like a computer program for calculating the mean of a set of numbers can be applied to any specific set of values. A typical Hierarchical

Task Analysis (HTA), (see Annett, Duncan, Stammers, & Gray, 1971; Kirwan & Ainsworth, 1992) results in a generative representation, in that the HTA chart can be followed to perform the task in any subsumed situation. The forms of GOMS models that explicitly represent methods (see John & Kieras, 1996a, 1996b) are also generative. The typical cognitive architecture model is generative in that it is programmed to perform the cognitive processes necessary to decide how to respond appropriately to any possible input that might occur in the task. In essence, the model programming expresses the general procedural knowledge required to perform the task, and the architecture, when executing this procedural knowledge, supplies all of the details; the result is that the model responds with a different specific time sequence of actions to different specific situations.

For example, Kieras, Wood, and Meyer (1997) used a cognitive architecture to construct a production-rule model of some of the telephone operator tasks studied by Gray, John, and Atwood (1993). Because the model consisted of a general "program" for doing the tasks, it would behave differently depending on the details of the input events; for example, greeting a customer differently depending on information on the display and punching function keys and entering data depending on what the customer says and requires. Thus, the specific behavior and its time course of the model depend on the specific inputs, in a way expressed by a single set of general procedures.

A generative model is typically more difficult to construct initially, but because it is not bound to a specific scenario, it can be directly applied to a large selection of scenarios to provide a comprehensive analysis of complex tasks. The technique is especially powerful if the model runs as a computer simulation in which there is (a) a *simulated device* that represents how the scenario data results in specific display events and governs how the system will respond to the user, and (b) the *simulated human*, which is the model of how the user will perform the task. The different scenarios are just the input data for the simulation, which produces the predicted behavior for each one. Furthermore, because generative models represent the procedural knowledge of the user explicitly, they readily satisfy the desirable property of models described above: The content of a generative model can be inspected to see how a design "works" and what procedures the user must know and execute.

The Role of Detail

In the initial presentation, the reader may have noticed the emphasis on the role of detailed description, both of the user's task and the proposed interface design. Modeling is sometimes criticized because it appears to be unduly labor intensive. Building a model and using it to obtain predictions may indeed involve substantial detail work; however, working out the details about the user's task and the interface design is, or should be, a necessary part of any interface design approach; the usability lies in the details, not the generalities (Whiteside, Jones, Levy, & Wixon, 1985). If the user's task has not been described in detail, chances are that the task analysis is inadequate and a successful interface will be more difficult to achieve; extra design iterations may be required to discover and correct deficiencies in the original understanding of the user's needs. If the interface designer has not worked out the interface design in detail, the prospects of success are especially poor. The final form of an interface reflects a mass of detailed design decisions. These decisions should have been explicitly made by an interface designer whose focus is on the user, rather than the programmers who happen to write the interface code. The designer must therefore develop this detail as part of any successful design effort. In short, using model-based evaluation does not require any more detail than should be available anyway; it just requires that this detail be developed more explicitly and perhaps earlier than is often the case.

Cognitive architectures are committed to detail. Cognitive architecture systems are primarily research systems dedicated to synthesizing and testing basic psychological theory. Because they have a heavy commitment to characterizing the human cognitive architecture in detail, they naturally work at an extremely detailed level. The current cognitive architecture systems differ widely in the extent to which they incorporate the most potent source of practical constraints—namely, perceptual-motor constraints—but at the same time, they are committed to enabling the representation of a comprehensive range of very complex cognitive processes, ranging from multitask performance to problem solving and learning. Thus, these systems are generally very flexible in what cognitive processes they can represent within their otherwise very constrained architectures.

The detail, however, has a downside. Cognitive architectures are typically difficult to program, even for simple tasks, and have the further drawback that, as a consequence of their detail, currently unresolved psychological issues can become exposed to the modeler for resolution. For example, the nature of visual short-term memory is rather poorly understood at this time, and no current architecture has an empirically sound representation of it. Using one of the current architectures to model a task in which visual short-term memory appears to be prominent might require many detailed assumptions about how it works and is used in the task, and these assumptions typically cannot be tested within the modeling project itself. One reason for this is the previously discussed difficulty of getting high-precision data for complex tasks. The more serious reason, though, is that in a design context, data to test the model is normally not available because there is not yet a system with which to collect the data. Less detailed modeling approaches such as GOMS may not be any more accurate, but they at least have the virtue of not sidetracking the modeler into time-consuming detailed guesswork or speculation about fundamental issues. See Kieras (2005b) for more discussion.

Task networks can be used before detailed design. Although model-based evaluation works best for detailed designs, the task network modeling techniques were developed to assist in design stages before detailed design, especially for complex military systems. For example, task network modeling was used to determine how many human operators would be required to properly man a new combat helicopter. Too many operators drastically increases the cost and size of the aircraft; too few means the helicopter could not be operated successfully or safely. Thus, questions at these stages of design are what

capacity (in terms of the number of people or machines) is needed to handle the workload, and what kinds of work needs to be performed by each person or machine.

In outline, these early design stages involve first selecting a *mission profile*, essentially a high-level scenario that describes what the system and its operators must accomplish in a typical mission, then developing a basic *functional analysis* that determines the functions (large-scale operations) that must be performed to accomplish the mission, and what their interactions and dependencies are. Then the candidate high-level design consists of a tentative *function allocation* to determine which human operator or machine will perform each function (see Beevis, Bost, Doering, Nordo, Oberman, Papin, et al., 1992). The task network model can then be set up to include the tasks and their dependencies, and simulations run to determine execution times and compute workload metrics based on the workload characteristics of each task.

Clearly, entertaining detailed designs for each operator's controls or workstation is pointless until such a high-level analysis determines how many operators there will be and for which tasks each is responsible. Note that the cognitive-architecture and GOMS models are inherently limited to predicting performance in detailed designs, because their basic logic is to use the exact sequence of activities required in a task to determine the sequence of primitive operations. However, as will be discussed below, recent work with *high-level GOMS models* suggests an alternative approach in which a GOMS model using abstract or high-level operators to interact with the device can be developed first, and then elaborated into a model for a specific interface as the design takes shape. However, at this time, for high-level design modeling, the task-network models appear to be the best, or only, choice.

The limitations, however, must be clearly understood. The ability of the task network models to represent a design at these earliest stages is a direct consequence of the fact that these modeling methods do not have any detailed mechanisms or constraints for representing human cognition and performance. Recall that the tasks in the network can consist of any arbitrary process whose execution characteristics can follow any desired distribution. Thus, the tasks and their parameters can be freely chosen without any regard to how a human will actually be able to do them in the final version of the system. Hence, this early-design capability is a result of a lack of theoretical content in the modeling system itself.

While the choice of tasks in a network model is based on a task analysis, the time distribution parameters are more problematic; how does one estimate the time required for a human to perform a process specified only in the most general terms? One way is to rely on empirical measurements of similar tasks performed in similar systems, but this requires that the new system must be similar to a previous system not only at the task-function level, but also at least roughly at the level of design details.

Given the difficulty of arriving at task parameter estimates rigorously, a commonly applied technique is to ask a subject matter expert to supply *subjective estimates* of task time means and standard deviations and workload parameters. When used in this way, a task-network model is essentially a mathematically straightforward way to start with estimates of individual subtask performance, with no restrictions on the origin or quality of these estimates, and then to combine them to arrive at performance estimates for the entire task and system.

Clearly, basing major design decisions on an aggregation of mere subjective estimates is hardly ideal, but as long as a detailed design or preexisting system is not available, there is really no alternative to guide early design. In the absence of such analyses, system developers would have to choose an early design based on "gut feel" about the entire design, which is surely more dangerous.

Note that if there is a detailed design available, the task-network modeler could decompose the task structure down to a fine enough level to make use of basic human performance parameters, similar to those used in the cognitive-architecture and GOMS models. For example, some commercial tools supply the HOS micromodels; however, it is hard to see the advantage in using task network models for detailed design. The networks and their supplementary executable code do not seem to be a superior way to represent task procedures compared to the computer-program-like format of some GOMS models, or the highly flexible and modular structure of production systems.

Another option would be to construct GOMS or cognitive architecture models to produce time estimates for the individual tasks, and use these in the network model instead of subjective estimates. This might be useful if only part of the design has been detailed, but otherwise, staying with a single modeling approach would surely be simpler. If one believes that interface usability is mostly a matter of getting the details right, along the lines originally argued by Whiteside, Jones, Levy, & Wixon (1985) and verified by many experiences in user testing, modeling approaches that naturally and conveniently work at a detailed design level will be especially valuable.

PRACTICAL ISSUES IN APPLYING A MODEL-BASED EVALUATION TECHNIQUE

Once a model-based evaluation technique is chosen, some practical issues arise in seeking to apply the technique to a particular user interface design situation. This section presents several of these issues.

Creating the Simulated Device

As previously mentioned, the basic structure of a model used for evaluation is that a simulated human representing the user is interacting with a simulated device that represents the system under design. In a parallel with Norman's interactive cycle, the simulated human receives simulated visual and auditory input from the simulated device, and responds with simulated actions that provide input to the simulated device, which can then respond with different visual and auditory inputs to the human. Depending on the level of generativity and fidelity of the model, the simulated device can range from being a dummy device that does nothing in response to the simulated human interaction, to a highly detailed simulation of the device interface and functionality. An example of a dummy device is the device that is assumed in the Keystroke-Level Model, which is not at all explicitly de-

fined; the modeler simply assumes that a specific sequence of user actions will result in the device doing the correct thing. At the other extreme are models such as the ones used by Kieras and Santoro (2004), which actually implemented significant portions of the logical functionality of a complex radar workstation and the domain of moving aircraft and ships. In modeling situations where a generative model is called for—namely, a complex task domain with multiple or lengthy detailed scenarios—a fully simulated device is the most convenient way to ensure that a simulated human is in fact performing the task correctly, and to easily work with more than one scenario for the task situation.

It is important to realize that the simulated device is not required to implement the actual interface whose design is being evaluated. Rather, it suffices to produce abstract psychological inputs to the simulated human. For example, if a red circle is supposed to appear on the screen, the simulated device can merely signal the simulated human with an abstract description that an object has appeared at certain (x, y) coordinates that has a "shape" of "circle," and a "color" of "red." It is not at all necessary for the simulated device to actually produce a human-viewable graphical display containing a circular red area at a certain position.

A lesson learned by Kieras and Santoro (2004) was that the effort required to construct even such an abstract simulated device in a complex domain is a major part of the modeling effort, and is more difficult in some ways than constructing the models of the simulated users. Clearly, to some extent, this effort is redundant with the effort required to develop the actual system, and so can undermine the rationale for modeling early the design and development process.

A common response to this problem is to seek to connect the cognitive architecture directly to an intact application or system prototype that plays the role of the simulated device, in short replacing the simulated device with an actual device. Work such as St. Amant and Reidl (2001) provides a pathway for interfacing to an intact application: The technique is basically to use the existing API of the application platform (e.g., Windows) to capture the screen bitmap and run visual object recognition algorithms on it to produce the description of the visual inputs to the simulated human, and outputs from the simulated human can be directly supplied to the platform to produce keyboard input or to control the cursor position. Even for the limited domain of Windows applications using the standard GUI objects, this is technically challenging, but quite feasible.

A less ambitious approach is instead of connecting to an intact application program, to instrument a prototype version of the interface, so that for example, when the prototype causes a certain object to appear on the screen, the simulated human is supplied with the visual input description. Given the considerable variety in how GUIs are implemented, this solution is not very general, but does have interesting solutions if the application prototype is programmed in Java, HTML, or similar cross-platform languages or general-purpose tools that can be used for prototyping. A good example is the HTML-based modeling tool described in John, Prevas, Salvucci, and Koedinger (2004).

Both of these methods of coupling a user model to an application, however, suffer from an easily overlooked limitation: The time when modeling is most useful is *early* in design, *before* the system has been prototyped. Thus, because coupling to a prototype or an application can only happen late in the development process or after development, these approaches come too late to provide the most benefit of model-based evaluation.

Multiple approaches for creating the simulated device are thus both possible and necessary. If the design questions can be answered with evaluation techniques such as the Keystroke-Level Model, then no simulated device is needed at all. If the model is for an existing application, coupling to the intact application is clearly the best solution. If a prototype is going to be constructed at this point in the design process anyway, using it as the simulated device is the best solution. But in the potentially most useful case, the simulated device must be created before any prototype or final application, making the fewest possible commitments to prototyping or coding effort; this requires constructing a simulated device from scratch, stripped down to the bare minimum necessary to allow a candidate design to interact with the simulated user. The next sections provide some advice on this process.

How to simplify the simulated device. Distinguish between device behavior that is relevant to the modeling effort and that which is not. Basically, if the simulated human will not use or respond to certain behaviors of the simulated device, then the simulated device does not need to produce those behaviors. A similar argument applies to the amount of detail in the behavior. Of course, as the interface design is elaborated, the simulated device may need to cover more aspects of the task. Good programming techniques will make it easy to extend the simulated device as needed; a good programmer is a definite asset to the project.

Distinguish between what the device has to provide to the simulated human, and what would be a convenience to the modeler. That is, while the device can be supplied with abstract descriptions, an actual graphical display of what the simulated device is displaying can be a very useful tool for the modeler in monitoring, debugging, or demonstrating the model. A very crude general-purpose display module that shows what the simulated human "sees" will suffice for these purposes, and can be reasonably easy to provide in a form that is reusable for a variety of simulation projects. However, developing this handy display should be recognized as an optional convenience, rather than an essential part of the simulated device.

Since programming the simulated device can be a significant programming effort, an attractive simplification would be a programming language that is specialized for describing abstract device behavior. Clearly, using such a language could be valuable if the modeling system already provides it and it is adequate for the purpose, especially if the device programming language can generates a prototype for the interface that can be directly coupled to the simulated human, moving the whole process along rapidly. An extensive project involving many different but similar interface designs would especially profit if the language matches the problem domain well.

However, in less than ideal situations a specialized language is unlikely to be an advantage. The reason is that to cover the full span of devices that might need to be simulated, the device programming language will have to include a full set of general programming language facilities. For example, to handle the Kieras and Santoro (2004) domain, trigonometric functions are

needed to calculate courses and trajectories, and containers of complex objects are required to keep track of the separate aircraft and their properties. Thus, specialized languages will inevitably have to include most of the same feature set as general-purpose programming languages, meaning that the developers of modeling systems will have to develop, document, maintain, and support—with editors and debuggers—a full-fledged programming language. This takes effort away from the functions that are unique to human performance modeling systems, such as ensuring that the psychology is correctly represented. In addition, the modeler will also have to expend the time and effort necessary to learn a specialized language whose complexity is similar to a general-purpose programming language, also taking effort away from unique aspects of the modeling effort.

A better choice would be to provide for the device to be programmed easily in a standard general-purpose programming language that modelers can (or should) know anyway, allowing reuse of not just the modeler's skills, but existing programming tools and education resources as well. A well-designed modeling system can ensure that a minimum of system-specific knowledge must be acquired before coding can begin.

Identifying the Task Strategy

A task analysis does not necessarily specify a task strategy. Human performance in a task is determined by (a) the logical requirements of the task, what the human is supposed to accomplish, as determined by a task analysis; (b) the human cognitive architecture, the basic mechanisms available to produce behavior; and (c) a specific strategy for doing the task, given the task requirements, the architecture, and what should be done in what order and at what time to complete the task. Thus, to construct a model for doing the task, one must first understand the task, then choose an architecture, and then choose a strategy to specify how the architecture will be used to accomplish the task. Identifying this strategy is the critical prerequisite for constructing a model.

Normal task analysis methods, such as those described in sources such as (Kirwan & Ainsworth, 1992; Beevis et al., 1992; Diaper & Stanton, 2004), do not necessarily identify the exact sequence of actions to perform with the interface under design, and they rarely specify the timing of actions. For example, anyone who has made coffee with a home coffee maker knows that there are certain constraints that must be met, but there is still considerable variation in the sequence and time in which the individual required steps could be performed. In fact, there can be variation on how the activity is organized; for example, one strategy is to use the visible state of the coffee maker as an external memory to determine which actions should be performed next (Larkin, 1989). A normal task analysis will not identify these variations. But even further, task analysis will not necessarily identify how any trade-offs should be decided, even as basic as speed versus accuracy, much less more global problems such as managing workload, dealing with multiple task priorities, and so forth.

Thus, to model a human performing in such situations, some additional information beyond a normal task analysis has to be added—namely, the specific task strategies that are used to accomplish the tasks. Conversely, the performance that a human can produce in a task can vary over a wide range, depending on the specific strategy that is used, and this is true over the range of tasks from elementary psychology laboratory tasks to highly complex real-world tasks (Kieras & Meyer, 2000). This raises a general problem: Given that we have a model that predicts performance on an interface design, how do we distinguish the effects of the interface design from the effects of the particular strategy for using the interface? Not only does this apply to the model performance, but also to the human's performance. It has always been clear that clever and experienced users could get a lot out of a poorly designed system, and even a reasonably well-designed powerful system could be seriously underused (Bhavnani & John, 1996). How can we predict performance without knowing the actual task strategy, and how does our model's task strategy relate to the actual user's task strategy?

Difficulties in identifying task strategy. The state of the art in cognitive modeling research for identifying a task strategy is to choose a candidate intuitively, build the model using the strategy, evaluate the goodness of fit to data, and then choose a better strategy and repeat until a satisfactory fit is obtained. If there is adequate detail in the data, such as the sequence of activities, it might be possible to make good initial guesses at the task strategy and then revise these through the modeling process. This iterative refinement process is known to be very slow, but more seriously, in system design, we normally do not have data to fit a model to; this is what the modeling is supposed to replace. The task strategy has to be chosen in the absence of such data.

Another approach is to get the task strategy by knowledge engineering techniques with existing task performers or other sources such as training materials. As will be argued below, it is especially important to identify the best (or at least a good) strategy for doing the task. However, a good strategy for doing a task, is often not obvious, even to experts. Even highly experienced people do not always know or use the best procedures; even the trainers may not know them, and it is common to discover that procedural training materials present suboptimal methods. Finally, and again most importantly, if the system is new, there are no experts or training materials to consult to see how it is used.

A heuristic: Model what users should do. Given the obstacles to identifying task strategies, how do we find out what strategies users will follow in using the system under development? The short answer is that it is too hard to find out within the constraints of a design process. Instead, start from the design goals that the system is supposed to meet, and assume that *users will be using the system as it is supposed to be used*. For example, if the system provides a feature that is supposed to allow the user to easily perform a certain task, assume that the simulated user will use that feature in the intended fashion. This is essentially a best-case analysis of the ability of the user to make use of the interface design. If the usability under these conditions is too low, then it would certainly be inadequate when used by actual users. It is a separate issue whether the users can or will use the system in this intended way, and the failures can be dramatic and serious (Bhavnani & John, 1996). Whether users use a system in

the intended way depends on several factors: (a) problems in the learnability of the design, which some models (see John & Kieras, 1996a, 1996b) can predict; (b) the quality of the training materials, which also can be improved with modeling (see John & Kieras, 1996a, 1996b); and (c) perhaps most importantly, matters beyond the scope of model-based evaluation, such as whether users have the opportunity or incentive to take full advantage of the system. Finally, there is no point to trying to improve a system under the assumption that users will ignore the capabilities that it provides. Thus, in terms of choosing a task strategy to model for design evaluation, the most effective approach is to assume that the design will be used *as intended.* Not only will this strategy be the easiest to identify and implement, but it is also most directly relevant to evaluating the design.

Within this basic strategy, there is another range of variation, which is whether the user attempts to perform at the highest possible level, or simply satisfices by performing at some adequate level. That is, people can use clever low-level strategies (what Gray & Boehm-Davis [2000] termed microstrategies), to greatly improve performance. For example, if the task is to classify "blips" on a radar display, the user can speed up considerably by looking for the next blip to inspect while still hitting the keys to respond to the previous one (Kieras, Meyer, & Ballas, 2001). On the other hand, the user is performing reasonably if the user finishes each blip before going on to the next one. Kieras and Meyer (2000) pointed out that in a variety of even elementary laboratory tasks, subjects do not always adopt high-performance strategies, even when large improvements would result; they are optional, not mandatory. Therefore, even if we are willing to assume that the design will be used as intended, how do we know whether the actual users will be going "all out" with a clever strategy versus just "getting the job done?" Again, the short answer is that it is too difficult to find out, especially in a design process where the system does not yet exist.

In response to this quandary, Kieras and Meyer (2000) proposed the *bracketing heuristic*. Construct a base model in which the user performs the task in a straightforward way using the interface as designed, but without any special strategic optimizations, a *slowest-reasonable* model. Derive from this a *fastest-possible* model that performs the task at the maximum performance level allowed by the cognitive architecture used for the model. The two models should bracket the actual user's future performance. If both models produce adequate performance, then the design should be adequate; if both produce inadequate performance, then the design needs to be improved. If the slowest-reasonable model is inadequate, but the fastest-possible model is acceptable, boosting the level of training or perhaps motivation of the user might result in satisfactory performance, although clearly improving the design would be a more robust solution.

Concerns Over Model Validity

Can you believe the model? Suppose a model implies critical design choices. Should you follow them? A poor response is to build and test prototypes just as if no modeling had been done. It could be argued that the modeling might have clarified the situation, but the purpose of model-based evaluation is to reduce the amount of prototyping and user testing required to refine a design. This response under-utilizes the approach.

A better response to the situation is to understand how the model implies the design choices by answering the question, "What aspects of the model are contributing to the outcome?" This can be done by profiling the model processing and analyzing the model structure. If the critical aspects of the model are known to be valid and appear to be properly represented, then the model results should be accepted. For example, perhaps one design is slower than the other simply because it turns out that more navigation through menus is required. The model processes involved are relatively simple and adequately validated in the literature; however, if the relevant aspects of the model are problematic, the result needs further study. For example, suppose the model for the better of two designs differs from the poorer design in assuming that the user can remember all of the information about previously inspected screen objects, and so does not need to search the screen again when the information is needed later. Because the bounds on visual memory are unclear, as discussed above, the modeling architecture might not enforce any bounds. Thus, the modeling result is suspicious for this reason, and the bounds might be much smaller than the model assumes. The modeler could then perform a sensitivity analysis to reveal how much the design choice might be affected by the problematic assumption. For example, the model could be modified to vary the number of previous inspected objects that could be remembered. At what value does this parameter change which design is predicted to be better? If the decision is not very sensitive to this parameter (e.g., because the effects are minor compared to the total time involved, or there are other improvements in the better design), then choosing the better design is a reasonably safe decision.

If the decision turns out to be sensitive to the problematic model assumption, then the situation becomes quite difficult. One possible solution is to remove the problematic aspect of the model by changing it in the direction of less human capability. This is both a conservative strategy and might be appropriate if the user will be under stress as well. If data or theory is available that resolves the issue, however, the modeler can go beyond the normal model-based process and modify the model or architecture to incorporate the more accurate psychological information.

Should you validate the model? Remember that testing with real users must be done sometime in the development process, because the models do not cover all of the design issues, and are only approximate where they do apply and, like any analytic method, can be misapplied. The model can thus be validated after use by comparing the final user test results to the model predictions. This will reveal problems in the accuracy of the model and its application to the design, and any modeling mistakes and design errors can then be corrected for the future.

However, should special data to validate the model be collected *prior* to using it to guide the design? While it would seem to be a good idea to validate a model before using it, the answer really should be *no* because validation is not supposed to be a normal part of using a predictive model of human performance. The whole idea of model-based evaluation is to avoid data collection during design. Only the developers of the modeling

methodology are supposed to be concerned about the validity, not the user of the methodology.

A couple of special cases about data collection need to be discussed. One is data collection to provide basic parameter values for modeling, such as how long it takes the user to input characters with a novel device. If the parameters concern low-level processes, the data collection is independent of a specific design and will be generally useful for many different modeling applications. The second special case is data collection to support modeling how an *existing* system is *actually* being used. Such a model cannot be constructed a priori, but rather, must be based on data about how actual users interact with the actual system. Due to the uncertainties involved in constructing a model based on human behavior, the model will have to be validated with a suitably complete and reliable data set before it can be taken as a usefully accurate model. This purpose of modeling is very different from the model-based evaluation approach presented in this chapter: instead of serving as a guide for designing a new system and a surrogate for user testing, the model is an explanation and characterization of observed behavior; it might serve as a guide for a new design, but only in the sense of characterizing the current situation that we want to improve upon. The model itself will not directly apply to the new design. In short, modeling the actual use of an existing system has very different methods, goals, and applications from modeling the usage of a system being designed.

If in spite of all of the above considerations, the validity of the model is in question, then it needs to be realized that a data set adequate for a scientifically sound test of validity must be much more controlled and detailed than normal user testing data, and can be very difficult to collect in the context of a development project. Furthermore, if the resources are available to do such data collection and analysis before the final stages of development, what function is served by modeling? If the model validity would be in doubt, couldn't the data collection resources be better devoted to user testing?

To elaborate on the difficulty of collecting adequate validation data, while all would agree that a model is almost certainly incorrect at some level, it is often mistakenly assumed that collecting empirical data on human performance in complex tasks is a royal road to certainty. Rather, as pointed out in Kieras and Santoro (2004), complex real-world tasks involve subtle user strategies, team interactions, influences of background knowledge, and the specifics of the scenarios. Such experiments are extremely slow and expensive to conduct, even with small samples, where the reliability of the results then comes into question. Clearly, it is not practical to run experiments using (a) many scenarios, (b) every reasonable design variation, (c) every candidate team organization, and (d) ample numbers of subjects. Furthermore, even for a well-chosen subset of these possibilities, it may be difficult to understand why people did what they did in the tasks; asking them is usually ambiguous at best, and their strategies might be idiosyncratic. Thus, the reliability, generalizability, and even the meaning of the data can be difficult to determine. In fact, it can be difficult to ensure that the model and the experiment are even relevant to each other. For example (see Kieras & Santoro, 2004), if the model is based on what users *should* do in the task and the test users don't follow the strategy that the model follows, then the failure of the model to behave the same way as the test users is actually irrelevant; the data is "wrong," not the model! Thus, even if deemed appropriate, attempting to validate a model of a complex task is likely to be impractically difficult for a normal development process.

Summary: Assessing model validity. Instead of collecting data to validate the model, assess its validity in terms of whether it meets the following basic requirements:

1. Is the model strategy based on an analysis of what users *should* do? If not, it is a poor choice to inform the design of a new system.
2. Is it likely that users can or will follow the same strategy as the model? If not, then the model is irrelevant. Either the model was misconstructed or the design is fundamentally wrong.
3. Are the assumptions about human abilities in the model plausible? If not, see the earlier suggestions.

If the answer to all three questions is yes, then the model results can be accepted as useful guidance for the design decisions without special validation efforts. Of course, it needs to be kept in mind that the model might be seriously incorrect, but modeling is not supposed to be perfect; it suffices merely that it help a design process.

GOMS MODELS: A READY-TO-USE APPROACH

As summarized above, GOMS is an approach to describing the knowledge of procedures that a user must have in order to operate a system. The different types of GOMS models differ in the specifics of how the methods and sequences of operators are represented. The aforementioned CPM-GOMS model represents a specific sequence of activity in terms of the cognitive, perceptual, and motor operators performed in the context of a simple model of human information processing. At the other extreme of detail, the Keystroke-Level Model (Card, Moran, & Newell, 1980) is likewise based on a specific sequence of activities, but these are limited to the overt *keystroke-level* operators (e.g., easily observable actions at the level of keystrokes, mouse moves, finding something on the screen, turning a page, etc.). The task execution time can be predicted by simply looking up a standardized time estimate for each operator and then summing the times. The Keystroke-Level Model has a long string of successes to its credit (see John & Kieras, 1996a). Without a doubt, if the design question involves which alternative design is faster in simple situations, there is no excuse for measuring or guessing when a few simple calculations will produce a usefully accurate answer.

It is easy to generalize the Keystroke-Level Model to apply to more than one specific sequence of operators. For example, if the scenario calls for typing in some variable number of strings of text, the model can be parameterized by the number of strings and their length. However, if the situation calls for complex branching or iteration and clearly involves some kind of hierarchy of task procedures, such sequence-based models become quite awkward and a more generative form of model is required.

The generative forms of GOMS models are those in which the procedural knowledge is represented in a form resembling an ordinary computer programming language, and are written

in a fairly general sort of way. This form of GOMS model can be applied to many conventional desktop computing interface design situations. It was originally presented in Card, Moran, and Newell (1983, chapter 5), and further developed by Kieras, Polson, and Bovair (Kieras & Polson, 1985; Polson, 1987; Bovair, Kieras, & Polson, 1990), who provided a translation between GOMS models and the production-rule representations popular in several cognitive architectures and demonstrated how these models could be used to predict learning and execution times. Kieras (1988, 1997) proposed a structured-natural-language notation, NGOMSL ("Natural" GOMS Language), which preserved the empirical content of the production-rule representation, but resembled a conventional procedural programming language. This notation was later formalized into a fully executable form, GOMSL (GOMS Language), for use in computer simulation tools that implement a simplified cognitive architecture that incorporates a simple hierarchical-sequential flow of control (Kieras, Wood, Abotel, & Hornof, 1995; Kieras, 2005a). This tool has been applied to modeling team tasks (e.g., Kieras & Santoro, 2004), and extended to provide analysis of error recovery methods supported by error-source heuristics (Wood, 2000). See Baumeister, John, and Byrne (2000) for a survey of computer tools for GOMS modeling.

Continuing the analogy with conventional computer programming languages, in generative GOMS models, the operators are like the primitive operations in a programming language; methods are like functions or subroutines that are called to accomplish a particular goal, with individual steps or statements containing the operators, which are executed one at a time, as in a conventional programming language. Methods can assert a subgoal, which amounts to a call of a submethod, in a conventional hierarchical flow of control. When procedural knowledge is represented explicitly in this way, and in a format that enforces a uniform "grain size" of the operators and steps in a method, then there are characteristics of the representation that relate to usability metrics in straightforward ways.

For example, the collection of methods represents "how to use the system" to accomplish goals. If a system requires a large number of lengthy methods, then it will be hard to learn; there is literally more required knowledge than for a system with a smaller number or simpler methods. If the methods for similar goals are similar—or, in fact, the same method can be used to accomplish different, but similar, goals—then the system is "consistent" in a certain, easily characterized sense. In a procedurally consistent system, fewer methods, or unique steps in methods, must be learned to cover a set of goals compared to an inconsistent system, and so it is easier to learn. One can literally count the amount of overlap between the methods to measure procedural consistency.

Finally, by starting with a goal and the information about the specific situation, one can follow the sequence of operators specified by the methods and sub-methods to accomplish the goal. This generates the sequence of operators required to accomplish the goal under that specific situation; if the methods were written to be adequately general, they should suffice to generate the correct sequence of operators for any relevant task situation. The times for the operators in the trace can be summed, as in the Keystroke-Level Model, to obtain a predicted execution time. Details of the timing can be examined, or "profiled," to see where the processing bottlenecks are.

Why GOMS Models Work

The reasons why GOMS Models have useful predictive and heuristic power in interface design can be summarized under three principles: The *rationality principle* (cf. Card, Moran, & Newell, 1983) asserts that humans attempt to be efficient given the constraints on their knowledge, ability, and the task situation. Generally, when people attempt to accomplish a goal with a computer system, they do not engage in behavior that they know is irrelevant or superfluous; they are focused on getting the job done. Although they might perform suboptimally due to poor training (see Bhavnani & John, 1996), they generally try to work as efficiently as they know how, given the system they are working with. How they accomplish a goal depends on the design of the system and its interface. For example, in a word-processing system, there are only a certain number of sensible ways to delete a word, and the user has some basis for choosing between these that minimizes effort along some dimension. Between these two sets of constraints—the user's desire to get the job done easily and efficiently, and the computer system's design—there is considerable constraint on the possible user actions. This means that we can predict user behavior and performance at a useful level of accuracy *just from the design of the system and an analysis of the user's task goals and situation*. A GOMS model is one way of combining this information to produce predicted performance.

Procedural primacy is the claim that regardless of what else is involved in using a system, at some level, the user must infer, learn, and execute procedures in order to accomplish goals using the system. That is, computers are not used purely passively—the user has to do something with them, and this activity takes the form of a procedure that the user must acquire and execute. Note that even display-only systems still require some procedural knowledge for visual search. For example, making use of the flight status displays at an airport requires choosing and following some procedure for finding one's flight and extracting the desired information, because different airlines use different display organizations, some of which are probably more usable than others. Because the user must always acquire and follow procedures, the complexity of the procedures entailed by an interface design is therefore related to the difficulty of using the interface. While other aspects of usability are important, the procedural aspect is always present. Therefore, analyzing the procedural requirements of an interface design with a technique such as GOMS will provide critical information on the usability of the design.

Explicit representation refers to the fact that any attempt to assess something benefits from being explicit and clear and relying on some form of written formalized expression. Thus, all task analysis techniques (Kirwan & Ainsworth, 1992; Beevis et al., 1992; Diaper & Stanton, 2004) involve some way to express aspects of a user's task. Likewise, capturing the procedural implications of an interface design will benefit from representing the procedures explicitly in a form that allows them to be inspected and manipulated. Hence, GOMS models involve writing out user procedures in a complete, accurate, and detailed form. By doing so in a specified format, it becomes possible to define metrics over the representation (e.g., counting the number of statements) that can be calibrated against empirical measurements to provide predictions of usability. Moreover, by making

user procedures explicit, the designer can then apply the same kinds of intuition and heuristics used in the design of software: Clumsy, convoluted, inconsistent, and "ugly" user procedures can often be spotted and corrected just like poorly written computer code. Thus, by writing out user procedures in a notation like GOMS, the designer can often detect and correct usability problems without even performing the calculations. This approach can be applied immediately after a task analysis to help choose the functionality behind the interface, as well as to help in the initial design decisions (Kieras, 2004). Work in progress consists of adding high-level operators to a GOMSL and GLEAN (see Kieras, 2005b) to directly support a "seamless" transition of computational modeling from the task and functionality level of analysis down to detailed design.

Limitations of GOMS Models

GOMS models address only the procedural aspects of a computer interface design. This means that they do not address a variety of nonprocedural aspects of usability, such as the readability of displayed text, the discriminability of color codes, or memorability of command strings. Fortunately, these properties of usability are directly addressed by standard methods in human factors.

Within the procedural aspect, user activity can be divided into the open-ended "creative" parts of the task (such as composing the content of a document, or thinking of the concept for the design of an electronic circuit on the one hand) and the routine parts of the task on the other (which consist of simply manipulating the computer to accept the information that the user has created), and then to supply new information that the user needs. For example, the creator of a document has to input specific strings of words in the computer, rearrange them, format them, spell check them, and then print them out. The creator of an electronic device design has to specify the circuit and its components to a CAD system and then obtain measures of its performance. If the user is reasonably skilled, these activities take the form of executing routine procedures involving little or no creativity.

The bulk of time spent working with a computer is in this routine activity, and the goal of computer system design should be to minimize the difficulty and time cost of this routine activity so as to free up time and energy for the creative activity. GOMS models are easy to construct for the routine parts of a task, because, as described above, the user's procedures are constrained by the task requirements and the design of the system, and these models can then be used to improve the ability of the system to support the user. However, the creative parts of task activity are purely cognitive tasks, and, as previously discussed, attempting to formulate a GOMS model for them is highly speculative at best, and would generally be impractical. Applying GOMS thus takes some task analysis skill to identify and separate the creative and routine procedural portions of the user's overall task situation.

Finally, it is important to recognize that while a GOMS model is often a useful way to express the results of a task analysis, similar to the popular Hierarchical Task Analysis technique (Annett et al., 1971; Kirwan & Ainsworth, 1992), building a GOMS model does not "do" a task analysis. The designer must first engage in task analysis work to understand the user's task before a GOMS model for the task can be constructed. In particular, identifying the top-level goals of the user and selecting relevant task scenarios are all logically prior to constructing a GOMS model.

CONCLUDING RECOMMENDATIONS

- If you need to predict the performance of a system prior to detailed design when overall system structure and functions are being considered, use a task network model.
- If you are developing a detailed design and want immediate intuitive feedback on how well it supports the user's tasks, write out and inspect a high-level or informal GOMS model for the user procedures while you are making the design decisions.
- If your design criterion is the execution speed for a discrete selected task, use a Keystroke-Level model.
- If your design criteria include the learnability, consistency, or execution speed of a whole set of task procedures, use a generative GOMS model such as CMN-GOMS or NGOMSL. If numerous or complex task scenarios are involved, use a GOMS model simulation system.
- If the design issues hinge on understanding detailed or subtle interactions of human cognitive, perceptual, and motor processing and their effect on execution speed, and only a few scenarios need to be analyzed, use a CPM-GOMS model.
- If the resources for a research-level activity are available, and a detailed analysis is needed of the cognitive, perceptual, and motor interactions for a complex task or many task scenarios, use a model built with the simplest cognitive architecture that incorporates the relevant scientific phenomena.

References

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Annett, J., Duncan, K. D., Stammers, R. B., & Gray, M. J. (1971). *Task analysis*. London: Her Majesty's Stationery Office.

Baumeister, L. K., John, B. E., & Byrne, M. D. (2000). A comparison of tools for building GOMS models. *Proceedings of CHI 2000* (pp. 502–509). New York: ACM.

Bhavnani, S. K., & John, B. E. (1996). Exploring the unrealized potential of computer-aided drafting. *Proceedings of the CHI'96 Conference on Human Factors in Computing Systems* (pp. 332–339), New York: ACM.

Beevis, D., Bost, R., Doering, B., Nordo, E., Oberman, F., Papin, J.-P., et al. (1992). *Analysis techniques for man-machine system design*. (Report AC/243(P8)TR/7). Brussels, Belgium: Defense Research Group, NATO HQ.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*, 1–48.

Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396–410.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Chubb, G. P. (1981). SAINT, a digital simulation language for the study of manned systems. In J. Moraal, & K. F. Kraas (Eds.), *Manned system design* (pp. 153–179). New York: Plenum.

Diaper, D., & Stanton, N. A. (Eds.) (2004). *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.

Elkind, J. I., Card, S. K., Hochberg, J., & Huey, B. M. (Eds.) (1989). *Human performance models for computer-aided engineering*. Committee on Human Factors, National Research Council. Washington: National Academy Press.

Glenn, F. A., Zaklad, A. L., & Wherry, R. J. (1982). Human operator simulation in the cognitive domain. *Proceedings of the Human Factors Society* (pp. 964–969). Santa Monica, CA: Human Factors Society.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds Matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experiment Psychology: Applied, 6*(4), 322–335.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human-Computer Interaction, 8*(3), 237–209.

Harris, R. M., Iavecchia, H. P., & Bittner, A. C. (1988). Everything you always wanted to know about HOS micromodels but were afraid to ask. *Proceedings of the Human Factors Society* (pp. 1051–1055). Santa Monica, CA: Human Factors Soceity.

Harris, R., Iavecchia, H. P., & Dick, A. O. (1989). The Human Operator Simulator (HOS-IV). In G. R. McMillan, D. Beevis, E. Salas, M. H. Strub, R. Sutton, & L. Van Breda (Eds.), *Applications of human performance models to system design* (pp. 275–280). New York: Plenum Press.

John, B. E., & Kieras, D. E. (1996a). Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction, 3*, 287–319.

John, B. E., & Kieras, D. E. (1996b). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*, 320–351.

John, B., Prevas, K., Salvucci, D., & Koedinger, K. (2004). Predictive human performance modeling made easy. *Proceedings of CHI 2004* (pp. 455–462). New York: ACM.

John, B., Vera, A., Matessa, M., Freed, M., & Remington, R. (2002). Automating CPM-GOMS. *Proceedings of CHI 2002* (pp. 147–154). New York: ACM.

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *Handbook of Human–Computer Interaction* (pp. 135–158). Amsterdam: North–Holland Elsevier.

Kieras, D. E. (1997). A Guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 733–766). Amsterdam: North-Holland.

Kieras, D. E. (2004). Task analysis and the design of functionality. In A. Tucker (Ed.), *The computer science and engineering handbook* (2nd ed., pp. 46-1–46-25). Boca Raton: CRC.

Kieras, D. E. (2005a). *A guide to GOMS model usability evaluation using GOMSL and GLEAN4.* Retrieved March 26, 2007 from ftp://www.eecs.umich.edu/people/kieras

Kieras, D. (2005b, July). *Fidelity issues in cognitive architectures for HCI modeling: Be careful what you wish for.* Paper presented at the 11th International Conference on Human Computer Interaction (HCII 2005). Las Vegas, NV.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis in the application of predictive models of human performance. In J. M. C. Schraagen, S. E. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis* (pp. 237–260). Mahwah, NJ: Lawrence Erlbaum.

Kieras, D., Meyer, D., & Ballas, J. (2001). Towards demystification of direct manipulation: Cognitive modeling charts the gulf of execution. *Proceedings of CHI 2001* (pp. 128–135). New York: ACM.

Kieras, D. E. & Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies, 22,* 365–394.

Kieras. D., & Santoro, T. P. (2004). Computational GOMS modeling of a complex team task: Lessons learned. *Proceedings of CHI 2004: Human Factors in Computing Systems* (pp. 97–104). New York: ACM.

Kieras, D. E., Wood, S. D., Abotel, K., & Hornof, A. (1995, November). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. *Proceeding of UIST* (pp. 91–100). New York: ACM.

Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. ACM *Transactions on Computer-Human Interaction, 4*, 230–275.

Kirwan, B., & Ainsworth, L. K. (1992). *A guide to task analysis*. London: Taylor & Francis.

Landauer, T. (1995). *The trouble with computers: Usefulness, usability, and productivity.* Cambridge, MA: MIT Press.

Lane, N. E., Strieb, M. I., Glenn, F. A., & Wherry, R. J. (1981). The human operator simulator: An overview. In J. Moraal & K. F. Kraas, (Eds.), *Manned system design* (pp. 121–152). New York: Plenum.

Larkin, J. H. (1989) Display Based Problem Solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon*. Hillsdale, NJ: Erlbaum.

Laughery, K. R. (1989). Micro SAINT − A tool for modeling human performance in systems. In G. R. McMillan, D. Beevis, E. Salas, M. H. Strub, R. Sutton, & L. Van Breda (Eds.), *Applications of human performance models to system design* (pp. 219–230). New York: Plenum Press. Also available at http://www.maad.com

McMillan, G. R., Beevis, D., Salas, E., Strub, M. H., Sutton, R., & Van Breda, L. (1989). *Applications of human performance models to system design*. New York: Plenum Press.

Newell, A. *Unified theories of cognition.* (1990). Cambridge, MA: Harvard University Press.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman, & Draper, S. W. (Eds.), *User centered system design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Petrosky, H. (1985). *To engineer is human: The role of failure in successful design.* New York: St. Martin's Press.

Pew, R. W., Baron, S., Feehrer, C. E., & Miller, D. C. (1977, March). Critical review and analysis of performance models applicable to man-machine systems operation. (Technical Report No. 3446). Cambridge, MA: Bolt, Beranek and Newman, Inc.

Polson, P. G. (1987). A quantitative model of human-computer interaction. In J. M. Carroll (Ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction.* Cambridge, MA: Bradford, MIT Press.

St. Amant, R., & Riedl, M. O. (2001). A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies, 55*(1), 15–39.

Strieb, M. I., & Wherry, R. J. (1979). *An introduction to the human operator simulator*. (Technical Report 1400.02-D). Willow Grove, PA: Analytics Inc.

Vera, A. H., Howes, A., McCurdy, M., Lewis, R. L. (2004). A constraint satisfaction approach to predicting skilled interactive cognition. *Proceedings of the CHI 2004* (pp. 121–128). New York: ACM.

Wherry, R. J. (1976). The human operator simulator—HOS. In T. B. Sheridan, & G. Johannsen (Eds.), *Monitoring behavior and supervisory control* (pp. 283–293). New York: Plenum Press.

Whiteside, J., Jones, S., Levy, P. S., & Wixon, D. (1985). User performance with command, menu, and iconic interfaces. *Proceedings of CHI'85* (pp. 185–191). New York: ACM.

Wood, S. D. (2000). *Extending GOMS to Human Error and Applying it to Error-Tolerant Design*. Doctoral dissertation, University of Michigan.

Zachary, W., Santarelli, T., Ryder, J., & Stokes, J. (2000). *Developing a multi-tasking cognitive agent using the COGNET/IGEN integrative architecture*. (Technical Report No. 001004.9915). Lower Gwynedd, PA: CHI Systems, Inc. Also available at http://www.chiinc.com/

Part

VII

MANAGING HCI AND EMERGING ISSUES

·61·

TECHNOLOGY TRANSFER

Kevin M. Schofield
Microsoft Corporation

Introduction **1212**
The New Face of Technology Transfer **1212**
Internal Technology Transfer **1212**
Technology Transfer as a Social Process **1213**
Building Trust in Steps **1213**
Thinking for the Future **1214**
Applying the Approach to HCI Tech Transfer **1214**
External Technology Transfer **1215**
Know What You're Selling **1215**
Know What You're Not Selling **1215**
Know the People Involved **1215**
Crafting a Deal **1216**
Confidentiality and NDAs **1216**
Going to the Negotiating Table **1217**
Embracing the Relationship **1217**
Commonalities for Internal and External Technology Transfer **1218**
Intellectual Property 1218
It All Comes Down to the Relationship **1218**
References **1218**

INTRODUCTION

This chapter is intended to provide guidance on how to approach technology transfer of HCI-related research. Admittedly, there are many different perspectives one could take in looking at this often-difficult problem: researcher versus practitioner, industry versus government versus academia. There is also the extra, added dimension of whether the transfer is within the boundaries of a corporation, or across corporate lines.

In this chapter, we will discuss tech transfer both internal and external to a company, and then discuss commonalities across the two processes. Primarily we will look at it from the perspective of the researcher, because in practice the burden is more on the researcher to justify the transfer and to make it work. Practitioners, however, will also gain value from reading this, as it will help them to understand the role of the researcher in technology transfer.

While any technology transfer has its challenges, HCI tends to be a particularly difficult one. This is due to many factors, but the two main ones are that it is often more about abstract ideas than specific implementations, and because we in the HCI community are still fighting the (wrong) impression that HCI is an afterthought and not the "real meat" of research and development efforts.

THE NEW FACE OF TECHNOLOGY TRANSFER

The last 10 years have seen a dramatic boom-and-bust cycle in the technology industry that, for a number of reasons, has changed the face of business, commerce, and information flow. Probably the most earthshaking of those changes has been within the high-tech industry itself, as fitting for such a rabid consumer of its own technologies and ideologies.

The Internet, through its nearly-ubiquitous connectivity to information and other people, has reshaped organizational patterns and forged a brand-new set of relationships between researchers, designers, developers, manufacturers, marketers, and the other people involved in business and commerce. Those new organizations and relationships respect neither national nor corporate borders, and bring us one step closer to the "friction-free" economy that was trumpeted in the mid-1990s.

Technology transfer is not what it used to be, largely because research and development is also not what it used to be. As Friedman (2005) pointed out, the world has flattened, and increasingly research and development can be done anywhere, with anyone, and for anyone. In lockstep, the relationships between corporate and academic R&D have also evolved; there are fewer and smaller standalone corporate research labs as many companies have chosen to "outsource" their basic research to academia. Even those companies that continue to invest in basic research are realizing that with fewer barriers to communication and collaboration, an "open innovation" model of joint R&D, as suggested by Chesbrough (2003), often makes sense to maximize the impact of internally generated ideas as well as to take full advantage of the most relevant externally generated ones.

Government research, and funding, is also changing. Some of this is due to economic cycles; much is due to shifting political priorities. In Europe, for example, the Lisbon Agenda defines the European Union's desire to make Europe the most innovative economy in the world by 2009, and this drives their investment in research and development. In contrast, in the United States, despite a wide recognition that IT is driving economic advances, the major funding agencies (NSF, DARPA, NIST) have reduced their support for academic research in computer science—with a built-in assumption that private industry should and will take up the slack. Somewhere in the middle, many countries (e.g., Canada, Australia) have programs where academic researchers who are funded by private companies can apply to the government for matching funding.

What should we take away from this? That clearly there is no one model for technology transfer from research into industry—there exists a whole spectrum. As mentioned previously, some governments fund research while others don't, and some are in the middle. Likewise, some companies do their own basic research, while others outsource it, and some are in the middle. Some universities keep a death-grip on their faculty's intellectual property; others, such as University of Wisconsin and University of Waterloo, have a policy that IP belongs to the inventor. Many are in the middle.

Consequently, there is no "playbook" that will explain exactly how to transfer a research result. Both researchers and practitioners will need to be very flexible and willing to adjust to the situation in which they find themselves. Tech transfer is, in the end, a business, and the best way to succeed at it is to think like a businessperson.

INTERNAL TECHNOLOGY TRANSFER

I am constantly surprised by the number of people who believe that technology transfer is some sort of Rube Goldberg machine,[1] where technology is inserted in one end of the contraption, strange things happen in the middle that usually involve people in uncomfortable and contorted positions, and then magically it pops out on the other end. Countless books and articles have been written (Lesko, Nicolai, & Steve, 1998; Mock, Kenkeremath, & Janis, 1993) in an attempt to document the perfect mechanical process for technology transfer. And yet, despite the fact that nearly everyone has had painful experiences trying to define a mechanical process for technology transfer, they still try to do it and complain bitterly when it fails (Butler, 1990; Hiltzik, 1999; Isaacs & Tang, 1999; Singh, 1993).

TECHNOLOGY TRANSFER AS A SOCIAL PROCESS

Tech transfer is not a mechanical or logistical process; it is fundamentally a social process. It succeeds when people build a relationship between the provider and the recipient that fosters trust and communication. Manning (1974) recognized that

[1]For those in the United Kingdom, that would be "Heath Robinson machine."

successful technology transfer centers on viewpoints and perspectives, and fundamentally on communications. Foley (1996) spoke to this point most directly—that technology transfer is a "full-contact sport" that centers on the people.

Successful product organizations understand that risk is their mortal enemy. They work hard to proactively manage the risk in their development process, or to remove the risk factors altogether. One of the most prevalent and difficult-to-manage risk factors is an external dependency, and let's be honest: an external dependency from a research organization looks about as risky as it gets. As long as your counterparts in product organizations think of a research organization that way, technology transfer is difficult at best and often outright impossible.

In order to succeed with technology transfer, we need to mitigate the risk, or at least the perceived risk in the minds of the people we wish to receive our technology.

Up to this point, none of this is particularly controversial, but this is where the paths diverge. Many people will tell you that you succeed in mitigating the risk by creating well-defined, step-by-step processes through which you and your industry partner will enact the technology transfer. I argue that this approach fails more often than it succeeds, for two main reasons:

- The only experience that people in industry have with external dependencies is the occasional dependence on an external contractor or supplier to deliver a finished component ready for integration. They inevitably use this as the model for defining their tech-transfer process from research, and it is fundamentally incompatible. Research technologies are not finished components, and any product organization that expects a research group to deliver a finished component fundamentally misunderstands the role, expertise, and hiring practices of a research organization. Research groups almost never understand the development and test practices of a commercial product organization; even if they did, those practices vary so widely between organizations that past experience does not guarantee that they could successfully deliver a finished component. Moreover, it is not the goal of a research organization to develop technologies into finished components; its goal is to discover and prove solutions to previously unsolved problems. To do so requires different skills and expertise, and, frankly, different development and testing methodologies. Both research and product organizations need to understand this fundamental difference, and embrace it as a way to complement each others' strengths, rather than ignore it and delude themselves about a theoretical capability that, practically speaking, is not there.
- People who don't understand each other can't communicate and don't trust each other, and can't be expected to work cooperatively toward a shared goal, even within the best-defined process. The trust and communication must come first. If the two people on opposite sides of a table trust each other, then the two of them can accomplish anything; if they don't, they will never accomplish anything of value.

BUILDING TRUST IN STEPS

So then comes the catch: how does one build trust? By working side by side, of course! This means that one must start with the kind of activities that are initially low in risk, but high in communication, and build on one's successes to build more trust and overcome successively higher levels of risk.

Step 1 is to establish trust that one is an expert in the domain. Technical people, whether in research or in industry, are almost universally avid readers and understand the importance of staying up to date in their fields. But we all suffer from a lack of time to weed through the volumes of less-than-useful information to find the truly valuable nuggets of wisdom. If someone in a product organization expresses an interest in one's field, an offer to forward them a set of papers, articles, and books that represent the seminal reading is a great first response. Granted, doing a literature search is not glamorous work, but it fundamentally demonstrates a working knowledge of the domain and an ability to provide guidance. Equally important, it shows a healthy respect for the people's intellect and a flattering assumption that they will be able to read and digest the material. One of two things will happen. Either they will actually read the materials sent to them, in which case they have not only made an initial investment in seeing a technology transfer happen but have also been provided with great topics for further conversation, or they will not read the material and most likely conclude that it is simply easier to rely upon the researcher as their expert rather than to become experts themselves. Either way, it is a foot in the door. They will ask endless questions as they try to decide for themselves what is within the realm of possibility and, more importantly, practicality. It is essential to ask them just as many questions to understand as completely as possible their constraints and to make clear recommendations on what they can expect to build.

Step 2 is to move from simply giving domain guidance on the state of the art, to providing feedback on product-design plans. This involves offering to review specifications and providing timely feedback. Timeliness is critical—schedules are the rules of the game, and an ability to stay within their stated schedule reflects an understanding of the rules, an appreciation for their importance, and a commitment to the success of their project. This is also a critical test of a researcher's ability to think practically; in their distrust, they might expect suggestions of wildly impractical things that would have a negative impact on their schedule or require resources out of proportion the relative importance of the technology to the overall product. It is the researcher's job to show once again an understanding of their constraints and the value added to their team effort. Success will be apparent when a subtle but important shift happens: instead of the researcher asking to review their design documents, they will ask the researcher to review them.

Step 3 is a significant one indeed: when the clients ask the researcher to help write the specification for the product. Don't expect this to happen until there has been a clear success at Step 2 that has established credibility. But when it does happen, the product organization is making a loud and unambiguous statement that they now think of the researcher as part of the product team. This is an enormous step for a product group to take in their relationship, and it is a heavy responsibility to take on. At first, they will probably only delegate small parts, and often they might ask the researcher to cowrite design documents. But regardless of the size of the assignment, the key to success is the same: whatever is designed must be easily buildable and testable. If there is any significant disagreement on whether the design can be built or tested, the product organization will not

take the risk. Development organizations (at least the successful ones) are inherently conservative and will overstate the costs to build new technologies.[2] This is not only another test of whether the researcher understands their constraints, but equally whether he or she understands their development process. I encourage "aiming low" initially, and looking for indications from the team that they would like to work together to design something more aggressive. If there is success in codesigning components, they will loosen the reins and delegate more responsibility (with more autonomy).

Step 4 is where one (finally) gets involved in implementation. It has taken enormous patience on the part of the researcher to get here, and there are still landmines everywhere. No two development organizations are alike; they all have different practices for creating, documenting, integrating, accepting, testing, and deploying new products. It is impossible to understand all of their processes, and it is very unlikely that they will all be written down; yet, every one of them is an opportunity to break form and cause a rift. I would encourage asking the group manager how they bring a new employee into the group and what training and mentoring that person would go through; further, see if there are opportunities to take advantage of such a process to help to get up to speed. If one has made it this far, the product team wants to see a success as much as the researcher does. Because one's success is the other's success, the product team will be very reasonable about doing things to help themselves be understood by the researcher, especially if it is clear to them that is the goal. All development groups fall in love with their own processes, and one earns their cooperation by showing equal respect for those processes, no matter how silly they might seem to an outsider.

The key to success in a development process is to realistically promise and over-deliver. Set rational expectations; they should think that you can carry your own weight, but not that you are God's gift to engineering. Promise metrics for work and for the technologies created that are achievable but not overly aggressive, and then exceed those metrics. By doing that, it is possible to fully gain their trust and move on to discuss with them more aggressive technology transfers.

It is important to note that in this "pyramid" of sorts that we are building with increasing levels of risk and corresponding trust, it is possible to peak at any level. For instance, if the researcher doesn't have the development skills to codevelop components with a real product team, then don't try to do it! By all measures, every step in this process can be considered technology transfer. Product organizations need knowledge, understanding, and ideas about technology just as much as they need finished technology components; they need to understand what can't be built just as much as what can. And, most importantly, they need researchers to tell them honestly what they are and are not capable of doing for them. Even without ever delivering a finished component to a product organization, one can still have a litany of technology-transfer successes for which the product organization will sing praises. It is more important to proceed in measured steps built upon past successes and build the trust and the lines of communication that will guarantee future successes.

THINKING FOR THE FUTURE

It is also important to be thinking to the future—to be thinking about what comes next. There is always a desire to simply throw a technology over the wall and then to move on to the next research project, but this is unrealistic. It never really works that way, and even if it did, "throwing it over the wall" would end the relationship and any opportunities for future technology transfer; the ongoing relationship after the transfer is an opportunity to carry on a dialogue about the next great technology.

From the product-organization perspective, there is rarely a "clean" way to integrate a component. The overwhelming majority of development work is revisions to existing products; very rarely are new products started. And revision work means that new components need to be integrated into an existing legacy framework; this usually requires development work on both sides of the integration to ensure the optimal match.

APPLYING THE APPROACH TO HCI TECH TRANSFER

As if this were not difficult enough, applying this approach to HCI-related technology transfer introduces its own challenges. One can read *The Psychology of Human-Computer Interaction* (Card, Moran, & Newell, 1983) and learn that at a very fundamental level a set of scientific principles holds very broadly. However, we in the HCI community have also learned that interactive systems must be designed within the context of a particular task and human, and that this very fact makes it tricky at best, and misleading or impossible at worst, to try to generalize specific designs to other contexts. Even with the best of intentions and the most thorough usability testing, there are no clear guidelines about how much of an HCI-related research technology can actually be transferred, and particularly for integration into an existing product. So "throwing over the wall" is especially difficult, as it calls for potential redesign, as well as further development and integration, test, localization, support, and operations (it is increasingly a service world, after all).

In the traditional view of technology transfer, lack of a clean handoff would be fatal. In the "relationship" view, however, this is an opportunity to build a working model that lets you overcome the challenges and work side-by-side with a product organization to guide the transfer of your work.

Beyond simply moving up the pyramid, one can do other important things to deepen the relationship. Scheduling regular "maintenance" conversations can help to maintain communication channels and to keep abreast of activity in the product organization; it is also an opportunity to continue to update them on progress on new work. The relationship can also be used to improve the researcher's own work by learning about critical real-world issues. Good product organizations have a wealth of information about their customers; by using their access to real

[2]Ironically, it has been my experience that development organizations tend to estimate incorrectly not in the new technologies, but rather in the incremental improvements to legacy components, and particularly in the "integration" work in making multiple components work together.

customers (and aggregate information about them), the relevance of research activities can be improved. It is an opportunity to ask key questions, learn about critical product strategic direction, conduct user studies, and find out what difficult HCI and technical issues are about to become critical issues for real customers. This is the golden opportunity to get out of the ivory tower.

EXTERNAL TECHNOLOGY TRANSFER

Technology transfers outside the boundaries of a legal entity, regardless of whether they originate in a government agency, academia, or an industrial lab, are almost by definition cleaner types of transfers. This means that one will be participating in a transaction involving the sale or licensing of technology, or contracting to provide some service, or both. So, the first order of business is negotiating the "deal."

KNOW WHAT YOU'RE SELLING

It is critical to know and understand what is being sold: outright ownership of intellectual property, or a license to it? Is it a complete solution that has been developed, or just pieces? Those pieces might include any or all of the following:

- User interface design;
- System specifications;
- A working prototype;
- An actual implementation, tested to some level of quality;
- The source code for a software implementation;
- Copyrights;
- Patents;
- Working time as a commitment to support ongoing productization;
- A running service that you host.

KNOW WHAT YOU'RE NOT SELLING

It is equally important to know what is not being sold. If one would like to continue this work, he or she will need to make sure to preserve rights and ownership to continue that work. Otherwise, one could very well put oneself out of business by selling complete ownership to a valuable asset, or by signing an exclusive license which precludes licenses with any other company.

KNOW THE PEOPLE INVOLVED

Deals inevitably involve lawyers as well as what are known as "business development" people: those whose job are to negotiate deals that further the business interests of their employers. One can assume that the company negotiating will have both business-development people and lawyers; it makes good sense for the researcher to have them too. This is a situation where we need to put our pride and high-minded notions about doing business "on a handshake" aside. While the overwhelming majority of companies are not in the business of stealing from people like us, and will not try to do so, in order to get the most value in return for what is being offered one needs someone on his or her side who understands what is customary in intellectual property deals and how to negotiate for it. Even a brilliant and excellent debater who doesn't know what to reasonably ask for is at a serious disadvantage. The bottom line: find someone with good business-development skills and experience to negotiate the deal.

Likewise, once the terms of a deal have been negotiated, one needs a lawyer to write it up and make it legally binding. Don't even think about self-representation in the drafting of an intellectual property agreement; the laws are changing too quickly (which isn't the fault of the lawyers) for a researcher to understand which ones apply to the situation and should be factored into the drafting of an agreement. Mock, Kenkeremath, and Janis (1993) set out the basics of existing laws and how they relate to technology transfer.

The key to success is to understand the defined role of each of the three people on a negotiating team: oneself, the business-development person, and the lawyer. The researcher's role is to be the technical expert and to place a value on the work as well as what the people on the other side of the table are offering in return; one's role as "client" is to decide what is needed in order to be successful, and what additionally is desired but negotiable. The role of the business-development person is to take the articulated needs and desires and try to structure the terms of an agreement that will work for both parties. He or she understands business risks and will help the team members to understand them and make informed decisions about how much risk can be tolerated. The role of the lawyer is to take the terms and write them down in words that both parties understand and that can be interpreted under the law to protect the client's interests. The lawyer also understands and can articulate the legal risks; laws are often subtle and ambiguous things that can be interpreted in many ways (in fact, nations have an entire branch of government that does nothing but interpret the laws). Any contractual obligation runs the risk of being interpreted in a way other than how it was intended, and a lawyer can help the team members to understand how likely that is, based upon the language of the law and similar previous cases where the law has been interpreted by the courts. Just as there is always business risk, there is always legal risk, and in the end it will fall upon the researcher to make the decision as to whether the risk is acceptable.

Where do deals go wrong? In my personal experience, they often go wrong when these three roles become confused, and when the business-development person and the lawyer start to make the key decisions. The researcher must live with the result of the deal, not them. On one end of the spectrum, there is no such thing as a risk-free deal; on the other end, even a high-risk deal could be worth doing if the reward is high enough. Those decisions are the client's, not theirs, and the researcher as client should insist on making them.

There are many good sources for both business-development people and lawyers. Venture capitalists will often have a

"short list" of ones they trust to do their business transactions. The Chamber of Commerce for a local area can also provide recommendations, and often will track complaints registered against specific ones. Many regions also have associations of entrepreneurs, inventors, and small-business owners, with great resources to draw from.

Of course, if one already works in a research lab, the institution most likely has a technology-licensing office that will negotiate and draft deals on the researcher's behalf (and are likely required to do so if the researcher's employment agreement assigns ownership of inventions to the employer). In that case, one will still need to stay involved to make sure that the researcher's needs are met in whatever deal is negotiated.

CRAFTING A DEAL

The most difficult process of negotiating a deal is crafting an arrangement that meets the needs of both sides. I have seen many negotiations take much longer to conclude, and in many cases fail to conclude successfully, because either or both sides didn't bother to try to understand the other side's business needs. Business partnerships are always about finding a way to help both parties be more successful. The best way to do that is to understand what one's prospective partner's business is about, and likewise to share enough information about one's own business, openly and honestly, so that together a combination can be found that works for both sides. Find out everything available about a partner's current business situation:

- their revenues and profits
- their competition
- their most important customers
- what customers are saying about their product
- where they say they want to take their business in the future
- the problems and challenges they are facing.

This essential information will guide you to deeper insights on how to offer terms that will be seen as valuable to a potential buyer or licensee. Steinberg (1998) described his experiences in negotiating deals and his own well-known and well-respected philosophy for how to structure deals makes good business sense for all parties.

Companies will pay for the value that can be delivered to them. They will pay a certain amount of money to make an even larger reduction in their costs (because in the end they save money). They will of course pay to help themselves make even more money. And, finally, they will pay if one solves a problem for them. As a deal is structured, try to cast it in terms of what it does for them; those are terms that they can understand and, more importantly, quantify in a valuation.

Take it further. Help the buyers in any way possible to place a value on what is brought to the table. For example, do a user study on their existing product, and another showing how the technology being offered will improve their product (if it happens to address a key customer complaint, all the better, and that should definitely be brought to their attention). During the negotiations, show them a smart, effective professional who can work with them. This doesn't mean that you need to negotiate hard to the very last item; contrary to popular belief, the tough negotiators aren't always the most respected, and in fact they are often the ones that create their own reputation for being difficult to work with. It is much more important to demonstrate an understanding of the buyers and an ability to speak their language, as well as willingness to help them make the case to the decision-makers in their organization's senior management.

CONFIDENTIALITY AND NDAS

One large challenge in negotiating deals is the issue of confidentiality, which usually rears its head first in the often-dreaded non-disclosure agreement. Non-disclosure agreements are signed prior to revealing confidential information, to ensure that the information won't be disclosed to competitors. That part is good business and a natural, noncontroversial part of good-faith bargaining—we all need to be able to keep secrets. The difficult part of non-disclosures is the issue of residuals: by looking at confidential information, I learn things, and then I carry that learning around in my head for the rest of my career. What am I allowed to do with that information in my head, and who in fact owns it? From the discloser's point of view, one wants to make sure that someone can't use his or her own confidential information to compete. From the other side, it is impossible to know exactly what will be disclosed, or what business opportunities are going to come one's way tomorrow, so it is deeply problematic to sign away the ability to enter certain businesses simply for the privilege of looking at confidential information. Both sides sound very reasonable, and they are, which is why NDAs are no trivial matter and often become the stopping point in negotiations.

Whenever possible, try to complete as much of the negotiations as possible without entering into an NDA, because it simplifies matters and prevents the trust issues from rising to the surface too early. The downside is that this makes the early negotiations a precarious dance, where a researcher needs to show the other party enough to convince them that the technology is real and solves their problem, without giving away key secrets. There are things that still can be done: tell them "what" it does, instead of "how" it does it, and show them the system working. The goal is to make them crave it enough that they will want to sign an NDA under the researcher's terms in order to complete the technical due-diligence required for them to close the deal.

It is critical to think this all through before getting to the negotiation table—these are never decisions to make under time and social pressure. It is also critical to realize that the whole issue of confidentiality and NDAs is one more business risk; admit it and decide for yourself whether (and when) the potential reward outweighs the risk. This is one of the clear cases in which a lawyer will be extremely conservative and protective, and describe in great detail everything that could be lost by entering into an NDA (or by showing technology without an NDA). But, in the end, the decision is the researcher's.

GOING TO THE NEGOTIATING TABLE

Negotiating a deal is probably the most hyped and feared part of this whole process; perhaps we've all had too many nightmares about slick car salesmen tricking us into paying too much for too little in return. The reason we fear car salesmen is that the salesman has all of the information about what the car is really worth and shares none of it with us; we are forced to blindly trust him, and many of us don't.

Steinberg (1998) once again shared his wisdom on a sound and ethical approach to negotiation in his "twelve essential rules for negotiating." Not nearly as ambitious as Steinberg, I have only three basic rules for negotiating:

1. I obtain as much information about each side's position as possible, before arriving at the table.
2. I have a list of what I really need in order to succeed, and a separate list of what I want to have in addition. I hold firm on my needs, and I am willing to compromise on my wants.
3. I always negotiate a deal in which both sides win.

Understanding a potential business partner's position is critical to negotiating success for a number of reasons. First, it tells what they are looking for. Ask the same list of questions we discussed with internal technology transfer: Who are their customers? What are those customers saying about their products? Who are their competitors? What is the company looking for that will help them to be more successful? What are their strengths and weaknesses? Second, and very much related, it tells how the buyer will value what they are offered. Successful business deals involve an exchange of value which both sides view as fair and equitable, but value is of course relative to the company and its context. Understand that in order to find an equal trade. Reading the company's annual report is an excellent source of information about a company (if it is a publicly traded company). Reading news articles and competitive reviews also provides invaluable information about the business pressures the company is under, as well as the assets that they bring to the table.

Having the list of the things that one really needs and the things that one wants in addition is a valuable step in preparing for negotiation. I have seen many people come to the table unwilling to compromise on anything; they ask for too much in the beginning, and believe strongly that compromising on anything is a sign of weakness. Negotiations like those always take longer than they should and are very frustrating. In some cases they spend more money in lawyers' fees for fruitless negotiations than the value of the small items on which they refuse to compromise. I recommend starting with a basic negotiation on the core needs of both sides; not only does that keep you focused on the heart of the deal, but it also tends to simplify things just by taking all the peripheral items off the table. Once the heart of the deal is done, and both sides feel comfortable that they can be successful because they are getting what they need, adding additional pieces is much easier with a lower stress level and a structure in place.

Remember to be honest with yourself as you detail all this information, because lying to yourself is the surest path to failure. Assume that they will have an accurate valuation of what is brought to the table (regardless of whether they are willing to tell you what it is), so insisting that something is worth more than its real value is foolish. Understand one's own strengths and weaknesses; be honest about what is needed in order to be successful, and don't promise things that can't be delivered.

All of this brings us to the last rule: always negotiate a win-win. Always negotiate a deal in which both sides feel that they are receiving what they needed and can be successful. Beyond the obvious ethical reasons for doing this, there is also the very practical consideration that the two parties will need to continue to work together. Many people go to the negotiating table believing that signing a deal is the end of the process, when in fact it is just the beginning. Once the paperwork is signed, a relationship that essentially lasts forever begins between the parties. This relationship often makes itself known in unanticipated ways; for example, if one is licensing a patent to a company, the company has a vested interest in ensuring that the patent maintenance fees continue to be paid to the PTO to keep the patent valid, and will want to have regular information to confirm that the payments are being made. Almost every deal one can imagine, no matter how cut-and-dry, has some aspect that will require communication between the parties on an ongoing basis after the deal is done. If one licenses a technology to a company, the company may be sending royalty checks, and the licensor may want some way to audit their sales to ensure that they are accurately paying. They, in turn, may want technical support, including important bug fixes and updates. They may additionally have negotiated the option to license future upgrades, and as a paying customer they will likely want to provide their input on features and enhancements to the technology that would be of most help to them.

EMBRACING THE RELATIONSHIP

By admitting from the beginning that there is an ongoing relationship, one can embrace this notion and turn it to an advantage. In fact, I encourage its use to build a future revenue stream—and in light of the increasing attention to Chesbrough's "open innovation" model, this is a likely outcome indeed. A researcher can build design and consulting services into the deal, which is particularly helpful for HCI-related technology transfer since, as we discussed earlier, they often need reworking to fit into a larger context. One can use the ongoing relationship as a "foot in the door" to be able to offer future sales and deals as new technologies are developed. In fact, viewed as a "strategic partner," one might even want to offer them the right of first refusal on future offerings, as a way of demonstrating a commitment to them. Finally, as in contemplating future growth of the researcher's business, one may need additional sources of funding, and a partner who has a vested interest in the researcher's success can be a great source for funding. Even if none of this is true, assume that one day there will be a need for a good reference or recommendation from them; that alone is reason enough to want to have a great ongoing relationship.

COMMONALITIES FOR INTERNAL AND EXTERNAL TECHNOLOGY TRANSFER

Intellectual Property

One of the issues that is common to both internal and external technology transfer is intellectual property (IP). I am not a lawyer, and so I obviously can't give legal advice on how to protect intellectual property or how to treat others' IP. What I can do is point out some places where the IP issues get thorny and make some business recommendations about how to deal with them.

Anything received from a third party may carry restrictions on how it may be used, and more importantly whether it can be redistributed in its original or modified form, or combined with some other components. This includes libraries of software routines, data, copyrighted works and designs, and patents. These restrictions can come from explicit license agreements that accompany the third-party components, or they could come from any of a number of different laws, including patent, copyright, trademark, trade secret, and export. Any time a third-party component is used in one's work, a business risk of constraining the ability to transfer the work to another party arises, because either one doesn't possess the right to do so, or the rights that are possessed aren't sufficient to the needs of the party that wants to license it. This question fits into the larger scheme of what it traditionally called the "build or buy" decision: whether it makes more business sense to build something oneself, or to buy or license it from a third party.[3]

I strongly recommend that, whenever possible, the IP issues should be dealt with at the time a third-party component is acquired, rather than waiting until an opportunity to transfer it. This accomplishes two things. First, it allows one to negotiate and make business decisions about acquiring the component before there is a commitment and dependence on the component built into your technology; once the dependency is there, the "switching cost" is much higher for moving to an alternate and one could be forced to pay a much larger licensing fee than before. Second, it simplifies the tech-transfer process. Any company worth its salt will perform a "due diligence" on the technology before it close a licensing deal. Part of that will be an analysis of who really owns the technology, or whether the licensor has acquired the right to further license it. In essence, one will need to prove the right to license one's work to the company. And don't be surprised if the company also asks the researcher to "warrant" the work—to guarantee that he or she has the right to license all of it to the company, and that right will be defended in court if necessary. The bottom line: clear it up front, and save a lot of trouble later.

IT ALL COMES DOWN TO THE RELATIONSHIP

The most significant common aspect of internal and external transfer comes back to the notion that we began with: technology transfer is a social process that succeeds or fails based upon the relationships that have been built. Tech-transfer partners need to trust each other, and that trust is built with communication and follow-through. Researchers should understand clearly and honestly what value they bring to the table at the various stages of the relationship, and they should make it their business to know how their partners see and value what they bring. After all, business is fundamentally the exchange of value between partners who need each others' competencies, and the truly successful companies are the ones that build relationships that last across a continuous series of business transactions.

When beginning a technology-transfer effort, assume that what is started that day is the beginning of a working relationship that will last forever. Build the relationship from the ground up with the expectation of ever-increasing levels of cooperation and trust that will allow the partnership to take on ever more challenging technology transfers, in whatever manner is most appropriate to the needs of the business.

There are far too many stories of companies who have struggled with the transfer of technologies that could change the world—and failed. As Buderi (2000) and Freidman (2005) described, the next chapter of this story is being written now. By rewriting the rules to focus on the social side of the process, we can ensure that our best work will see the light of day and this story will have a happy ending. The good news is that there are more companies open to technology transfer today than at any time in the history of the industry, and the opportunities are there—on both sides—for us to take.

References

Buderi, R. (2000). Engines of tomorrow: How the world's best companies are using their research labs to win the future. New York: Simon & Schuster.

Butler, K. (1990). Collaboration for technology transfer—or "How do so many promising ideas get lost?" *Proceedings of the CHI'90 Conference on Human Factors in Computing Systems* (pp. 349–351).

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale: Lawrence Erlbaum Associates.

Chesbrough, H. (2003). *Open innovation*. Boston: Harvard Business School Press.

Foley, J. (1996). Technology transfer from university to industry. *Communications of the ACM, 39*(9), 30–31.

[3]It is important to point out, though, that building from scratch doesn't necessarily mean an automatic escape from third parties; for example, one can still violate someone else's patent even with code written from scratch.

Friedman, T. (2005). *The world is flat*. New York: Farrar, Straus, and Giroux.

Hiltzik, M. (1999). *Dealers of lightning*. New York: Harper Collins Publishers.

Isaacs, E. A., Tang, J. C. (1996). Technology transfer: So much research, so few good products. *Proceedings of the CHI'96 Conference on Human Factors in Computing Systems*, (2) (pp. 155–156).

Lesko, J., Nicolai, P., & Steve, M. (1998). *Technology exchange in the information age*. Columbus, OH: Battelle Press.

Manning, G. K. (Ed.). (1974). *Technology transfer: Successes and failures*. San Francisco: San Francisco Press.

Mock, J. E., Kenkeremath, D. C., & Janis, F. T. (1993). *Moving R&D to the marketplace: A guidebook for technology transfer managers*. Falls Church: Technology Prospects, Inc.

Singh, G. (1993). From research prototypes to usable, useful systems: Lessons learned in the trenches. *Proceedings of the ACM Symposium on User Interface Software and Technology* (pp. 139–143).

Steinberg, L. (1998). *Winning with integrity*. New York: Villard Books.

·62·

AUGMENTING COGNITION IN HCI: 21ST CENTURY ADAPTIVE SYSTEM SCIENCE AND TECHNOLOGY

Dylan Schmorrow
Office of Naval Research

Leah Reeves
Potomac Institute for Policy Studies

Amy Kruse
Defense Advanced Research Projects Agency

Amy Bolton
Strategic Analysis

Introduction **1222**
Neurotechnologies for Assessing User State **1222**
Electrodermal Activity (EDA)/GSR-based Arousal and Cognitive Workload Gauge 1223
EKG (ECG)-Based Arousal and Cognitive Workload Gauges 1223
Body Position/Posture Tracking 1223
Stress Gauge 1226
Arousal Meter Gauge 1226
eXecutive Load Index Gauge 1226
P300 Novelty Detector Gauge 1227
Engagement Index Gauge 1227
New Workload Assessment Monitor (NuWAM; Combined EEG, ECG and EOG Sensors) 1227
Dry Electrodes 1227
Functional Near Infrared (fNIR) 1227
Boeing's fNIR Approach 1227
LM-ATL's fNIR Approach 1228
Pupillometry 1228
EEG 1229
LM-ATL's EEG-Based Technology 1229
DaimlerChrysler's EEG-Based Technology 1230
Event Related Optical Signal (EROS) 1230
Cognitive Monitor (CogMon) 1230
Application of Neurotechnologies **1230**
Closed-Loop Systems 1231
Honeywell's AugCog Closed-Loop Efforts 1231
DaimlerChrysler's AugCog Closed-Loop Efforts . . . 1231
LM-ATL's AugCog Closed-Loop Efforts 1232
Boeing's AugCog Closed-Loop Efforts 1232
QinetiQ's AugCog Closed-Loop Efforts 1232
Improving HCI System Design and Evaluation Capabilities 1233
Training Systems and Operator Selection 1233
CPE Initiative 1233
Assessing training progression 1234
Assessing individual potential 1234
Lessons Learned and Future Directions **1235**
The need to understand the user 1235
Integrate early, integrate often 1235
Development strategy 1235
One sensor/gauge does not fit all 1235
Cognitive bottlenecks may be task specific 1235
Artifact detection and correction 1235
Perceived stability/trust 1235
Timing of mitigation strategies 1236
Mitigation strategy principles 1236
Hardware integration 1236
Context gauges 1236
Individual differences 1236
Proactive vice reactive augmented cognition . . . 1236
Metrics 1236
Cognitive State Sensors 1237
Mitigation Strategies 1237
Robust Controllers 1237
Roles of National and Supra-National Institutions . . . 1237
Conclusion **1238**
References **1238**

INTRODUCTION

In 1960, J. C. R. Licklider had a vision for a "Man-Computer Symbiosis" in which the human and computer, while dissimilar from one another, would live together in an intimate association producing increased handling and new ways of processing information (Licklider, 1960). Over the past few decades, several attempts to realize this vision have been made by interactive system developers, but each time, it has eluded them. This was likely due to the insufficiency of technology and computational power, but also to the need to mature several fields of basic science necessary to understand how human-machine symbiosis might be produced. A more thorough understanding of human brain functioning and what guides behavior during human-computer interaction has been a continuing missing requirement in the ability to enable true human-machine symbiosis.

Fortunately, in the 1990s period known as the "Decade of the Brain," the National Institutes of Health (NIH) and other federal funding agencies invested heavily into advancing neuroscience in order to understand the basic scientific aspects underlying the brain to include human cognition and behavior. What emerged are advanced neurophysiological assessment tools and a wealth of knowledge regarding specific brain areas that could be attributed to particular cognitive and behavioral functions. As brain activity measurement tools, such as electroencephalography (EEG), magnetoencephalography (MEG), event related potentials (ERP), positron emission tomography (PET), and functional magnetic resonance imaging (fMRI), continued to advance into the year 2000, the idea that one could possibly begin to capture brain functioning in real time as users performed real world tasks came to fruition. Seeing the promise of such capabilities, in 2001, the Defense Advanced Research Projects Agency (DARPA) began funding a program based on what is now known as the scientific field of Augmented Cognition (AugCog) (Kollmorgen, 2007; Schmorrow & Kruse, 2002, 2004, 2005; Schmorrow & Reeves, 2007; Schmorrow, Stanney, Wilson, & Young, 2005). The field of AugCog has had many predecessors (e.g., DoD-funded programs in Biocybernetics, Learning Strategies, and the Pilot's Associate Program). It was not until the Decade of the Brain and after, however, that substantial investment in technologies to enable real-time monitoring of cognitive state via physiological sensors brought together a research community and stimulated the growth of the AugCog-based program that became known as the Improving Warfighter Information Intake Under Stress (IWIIUS) Program. The IWIIUS program included four industry teams working in different application domains toward a common goal of "closing the loop" around the human via the real-time assessment of human information processing (HIP) bottlenecks (e.g., sensory processing, attention, working memory, executive function) and use of subsequent adaptive automation techniques to mitigate the effects of those bottlenecks (Proffit, 2003). The ultimate was goal to improve human performance in both stationary (e.g., command and control) and mobile (e.g., in-vehicle, dismounted soldier) application domains.

The four main tenets of AugCog, on which the DARPA IWIIUS Program was founded, include using neuroscientific tools for assessing and accommodating the following HIP bottlenecks (Schmorrow, Stanney, Wilson, & Young, 2005):

- Sensory Processing: AugCog seeks to enhance sensory perception by exploiting multiple sensory channels for increased input capacity.
- Attention: AugCog seeks to equip computers such that they become aware of subtle cues emanating from humans indicating how they are prioritizing incoming information (e.g., directing attention) and capitalize on these cues to enhance HIP.
- Working Memory: AugCog seeks to support simultaneous processing of competing tasks by strategically allocating data streams to various multimodal sensory systems, while maintaining multimodal information demands within working memory capacity.
- Executive Function: AugCog seeks to enhance HIP by directing the recall of contextual information that cues the optimal interpretation of incoming information and moderates the effects of modality augmentation or switching.

Researchers and developers from AugCog and neuroergonomics (Parasuraman, 2003) efforts are aiming to revolutionize the field of Human-Computer Interaction (HCI) by leveraging recent advances in the fields of fields of neuroscience, cognitive science, cognitive neuroscience, and computer science and engineering to develop technologies and strategies to mitigate users' HIP limitations and improve their performance via real-time cognitive state (operator functional state, OFS) detection and accommodation. The AugCog science and technology (S&T) emerging from these research and development (R&D) efforts is providing tools and techniques that may be applied to (a) the design of systems (e.g., neurotechnologies as a measure during usability testing), (b) training (e.g., neurotechnologies to "sense" learning and systems to adapt training), and (c) operational systems (e.g., closing the loop between assessing cognitive state and initiating real time adaptive changes in the interface).

The aggressive goals and objectives required of the industry teams in the DARPA IWIIUS program (Schmorrow & Kruse, 2002, 2005; Schmorrow & McBride, 2004; Schmorrow, Stanney, Wilson, & Young, 2005) has fostered the development of many neurophysiological-based tools and techniques that are maturing enough to become feasible toolsets for HCI researchers, designers, and practitioners in their pursuit of improving human-computer system efficiency and effectiveness. This chapter is thus focused on summarizing some of the latest S&T advancements from the field of AugCog, with specific examples and lessons-learned from the DARPA program's four main industry teams (Honeywell Laboratories, Lockheed Martin Advanced Technology Laboratories [LM-ATL], DaimlerChrysler AG, Germany, and The Boeing Company) and their sub-contractors.

NEUROTECHNOLOGIES FOR ASSESSING USER STATE

The field of AugCog is aimed at substantially improving human-system interaction by using proactive systems that detect, gauge,

and assess a user's cognitive state in real time (via diagnostic physiological and neurophysiological sensors and appropriate data classification methods), and then mitigate HIP bottlenecks by modifying system interfaces (via dynamically adaptive strategies) (Schmorrow, Stanney, Wilson, & Young, 2005). A critical component of this goal is the use of neurotechnologies that can be used to appropriately measure cognitive load at a neural level in order to sufficiently characterize the cognitive state[1] of users while they interact with a computer-based system in real-world settings. Schmorrow and Kruse (2002) noted that there is a significant body of research—ranging from early measurements of pupil size to more recent observations using EEG, ERP, PET, and fMRI neurophysiological tools—that is devoted to measuring involuntary changes in human physiology as a result of cognitive load and effort (Gevins & Smith, 2000; Gundel & Wilson, 1992; Hess & Polt, 1964; Humphrey & Kramer, 1994; Just & Carpenter, 1993). Only a handful of these measurement techniques, however, have been used and analyzed in real time while users perform real-world tasks, and some devices (e.g., fMRI, PET) are still too expensive and big and thus not portable enough for use in operational environments.

HCI researchers and practitioners must be able to rely on tools and techniques that allow them easy access for noninvasively observing and assessing users while interacting with human-computer systems; therefore, only the physio- and neurotechnologies most appropriate for such assessment, and which have been proven by DARPA AugCog experiments in real-world settings, are reviewed here. For a more thorough review of sensor technologies available, readers are directed to the recent NATO report on Operator Functional State Assessment (Wilson & Schlegel, 2004).

In the last few years, AugCog researchers and developers have determined that various combinations of physiological (e.g., electro-oculography [EOG], heart rate [HR], galvanic skin response [GSR], blood pressure [BP], and pupillometry) and neurophysiological sensors may be used to improve the robustness for detecting cognitive functioning of users across a variety of conditions. The most promising to date are reviewed next and summarized in Table 62.1, and they include (a) stress and arousal meters; (b) engagement indices; (c) portable EEG, ECG (EKG), and functional near infrared (fNIR) devices; (d) posture/body state analyses; and (e) pupillometry. Each of the sensors/gauges discussed below were selected according to how effectively they have been shown by DARPA IWIIUS program performers to assess cognitive state in real time, while also considering factors such as portability, usability in the field, potential intrusion on user task performance, and potential electromagnetic interference (EMI) during combined sensor/gauge implementation in operational settings. (For detailed discussions of DARPA program performers' latest AugCog R&D efforts with various sensors and gauges in their various task environments, see proceedings articles from the recent First Annual Augmented Cognition International [ACI] Conference held in conjunction with HCI International 2005 in Las Vegas, Nevada.)

Electrodermal Activity (EDA)/GSR-based Arousal and Cognitive Workload Gauge

In pursuit of their DARPA AugCog-based IWIIUS program efforts, LM-ATL employed the use of an EDA/GSR-based arousal and cognitive workload gauge (Kobus, Brown, Morrison, Kollmorgen, & Cornwall, 2005). The system was provided by Anthrotronix, Inc. and included a device for capturing EDA/GSR data and the data processing algorithms necessary to convert its output into estimates of arousal and workload. In this system EDA is measured with a two-lead GSR sensor affixed to two nonadjacent toes on one foot of the user. Although GSR is typically measured using the palm of the hand or the fingers, the soles of the feet and toes have an equal abundance of eccrine glands that release sweat. Thus, the feet and toes offer an optimal site for sensor placement because they will not intrude on a user's ability to interact with input devices requiring hand operation (e.g., keyboard, mouse). The EDA (sampled at 32 Hz) data were processed to develop an arousal gauge, which was later integrated with other sensors to establish an effective multimeasure based general cognitive workload gauge.

EKG (ECG)-Based Arousal and Cognitive Workload Gauges

Anthrotronix, Inc. also provided LM-ATL with a tool used to collect EKG data, which included the data processing algorithms necessary to convert the EKG data into heart rate variability (HRV) and produce estimates of arousal and workload (Kobus et al.). To detect heart muscle activity, data were collected via three standard EKG electrodes, which were positioned in the center of the abdomen and below the left and right collarbone. Data from these EKG sensors (sampled at 256 Hz) was used to compute a user's HRV in milliseconds[2] each second in order to assess arousal states. LM-ATL and Anthrotronix also found that when concentrating on a task, the user's HRV decreases, and the heartbeat becomes very regular. Thus, HRV correlates with arousal may also serve as an indicator of cognitive workload. They also found that using HRV to estimate cognitive state for an individual required collecting only five minutes of baseline data prior to testing. Consequently, EKG-based arousal and cognitive workload gauges may be feasible tools for cognitive state assessment in most application settings.

Body Position/Posture Tracking

In support of multiple IWIIUS teams' AugCog efforts (e.g., LM-ATL, Boeing, DaimlerChrysler) the University of Pittsburgh provided a system for tracking body position and posture analysis (Kobus et al., 2005). In this system, data are collected via two main devices: a pressure-sensing chair and a head tracker. The sensor chair contains two 64×64 grids of pressure sensors

[1]In a forthcoming AugCog International Conference Proceedings publication (Reeves, Schmorrow, & Stanney, 2007), AugCog researchers and developers have defined cognitive or "functional" state to be: anything needing to be measured to base design decisions on it; the moment-to-moment dynamic and functional capabilities (e.g., capacity, bottlenecks) of the human brain, and; a condition that has a causative / moderating / predictive relationship to a performance variable.

TABLE 62.1. Summary of Maturing AugCog Technologies and Proven Implementations to Date*

Sensors/Gauges	Use (Measures What/Implemented How)	Appropriateness for Mobile Applications
EDA/GSR-based Arousal and Cognitive Workload Gauge (Anthrotronix)	*Provides estimates of arousal and general cognitive workload* • Implemented by LM-ATL in a command and control closed-loop application domain	Most appropriate for stationary users; not yet tested in mobile application domains or with mobile users
EKG (ECG)-based Arousal and Cognitive Workload Gauge (Anthrotronix)	*Uses* HRV *measures to provide estimates of arousal and general cognitive workload* • Implemented by LM-ATL in a command and control closed-loop application domain	Most appropriate for stationary users; not yet tested in mobile application domains or with mobile users
Body Position/Posture Tracking (U of Pittsburgh)	*Posture shift data, head position, and head velocity are used to gauge levels of attention (e.g., engagement)* • Implemented by LM-ATL and Boeing in 2 different command and control closed-loop application domains • Implemented by DaimlerChrysler in a vehicular closed-loop application domain	Appropriate for stationary users in stationary or mobile application domains; not yet tested with mobile users and most likely would not be appropriate for such
Stress Gauge (IHMC)	*Uses* VOG, HFQRS, *and* EDR *to track autonomic response to time-pressured, high workload tasks and to detect moment-to-moment cognitive stress related to managing multiple competing tasks and is thus good for measuring attention* HIP *bottleneck effects* • Implemented by Honeywell in closed-loop dismounted soldier application domains	Appropriate for mobile users and stationary or mobile application domains
Arousal Meter Gauge (Clemson)	*Uses* IBI *derived from* ECG *to track decrements in performance due to low arousal states in divided attention and vigilance tasks and is thus a good measure of attention* HIP *bottleneck effects* • Implemented by Honeywell in closed-loop dismounted soldier application domains • Implemented by Boeing in a command and control closed-loop application domain	Appropriate for mobile users and stationary or mobile application domains
eXecutive Load Index (XLI Gauge) (Human Bionics)	*Uses* EEG *measures to assess ability to allocate attentional resources during high workload, competing tasks and is thus a good measure of attention* HIP *bottleneck effects and general cognitive workload* • Implemented by Honeywell in closed-loop dismounted soldier application domains	Appropriate for mobile users and stationary or mobile application domains
P300 Novelty Detector Gauge (CCNY/Columbia)	*Uses* EEG *auditory* P300 *signals from frontal and parietal electrodes to track attentional resources used to attend to novel stimuli and is thus a good measure of attention* HIP *bottleneck effects* • Implemented by Honeywell in closed-loop dismounted soldier application domains	May be appropriate for mobile users and stationary or mobile application domains
Engagement Index Gauge (NASA/CCNY/Honeywell)	*Uses* EEG*-based measures to track how cognitively engaged a person is in a task (level of alertness) and is effective at assessing attention* HIP *bottleneck effects associated with both sustained and divided attention tasks, particularly during low workload conditions* • Implemented by Honeywell in closed-loop dismounted soldier application domains	May be appropriate for mobile users and stationary or mobile application domains
NuWAM combined EEG, ECG, EOG sensors (AFRL)	*Uses combined sensors to gauge general workload levels and estimate executive function and attention* HIP *bottleneck effects* • Implemented by Boeing in a command and control closed-loop application domain	Most appropriate for stationary users; not yet tested in mobile application domains or with mobile users
Fast fNIR device (Drexel)	*Measures brain blood oxygenation and volume changes and is an effective tool for assessing spatial and verbal working memory* HIP *bottleneck effects* • Implemented by LM-ATL in a command and control closed-loop application domain	Most appropriate for stationary users but shows promise for mobile users and mobile application domains

(*continues*)

TABLE 62.1. (*Continued*)

Sensors/Gauges	Use (Measures What/Implemented How)	Appropriateness for Mobile Applications
Whole Head fNIR (Archinoetics)	*Measures brain blood oxygenation and volume changes and is an effective tool for assessing spatial and verbal working memory* HIP *bottleneck effects* • Implemented by Boeing in a command and control closed-loop application domain	Most appropriate for stationary users; not yet tested in mobile application domains or with mobile users
Pupillometry (ETI's ICA)	*Uses proprietary and patented techniques for estimating cognitive activity based on changes in pupil dilation and gaze and is a good measure of general cognitive workload and sensory input, attention and executive function* HIP *bottleneck effects* • Implemented by LM-ATL and Boeing in 2 different command and control closed-loop application domains	Most appropriate for stationary users; not yet tested in mobile application domains or with mobile users
Low Density EEG (ABM's 3, 6, or 9 channel cap)	*Uses a portable* EEG *cap, wireless transmitter, and* B-Alert *software to effectively estimate various types of cognitive states, namely: vigilance/arousal, workload, engagement, distraction/drowsiness, and working memory levels* • Implemented by LM-ATL in a command and control closed-loop application domain	Appropriate for mobile users and stationary or mobile application domains
High density EEG (EGI's 128 or 256 electrode net)	*Uses an* ERP EEG*-based system to estimate which and to what degree particular brain regions are invoked during task performance; may be an effective tool for assessing both verbal and spatial working memory and general cognitive workload* • Evaluated but not implemented by LM-ATL in their command and control closed-loop application domain	Appropriate for stationary users; not yet tested in mobile application domains or with mobile users and may be too cumbersome for such applications
DaimlerChrysler's EEG system (FIRST)	*Uses* EEG *combined with* EOG *and* EMG *to assess low versus high workload levels and is effective at assessing sensory memory bottleneck effects* • Implemented by DaimlerChrysler in a vehicular closed-loop application domain	Appropriate for stationary users in stationary or mobile application domains; not yet tested on mobile users
EROS (U of Illinois)	*Uses fast optical imaging techniques to identify brain region signatures resulting from cued and non-cued attentional shifts during task performance and thus may be a good estimate of sensory, attention, and executive function* HIP *bottleneck effects* • Evaluated for potential implemented in Boeing's command and control closed-loop application domain	Appropriate for stationary users in stationary or mobile application domains; not yet tested with mobile users
CogMon (QinetiQ)	*Uses behavioral measures from interactions with cockpit controls,* EEG*-based physiological measures, subjective measures, and contextual information to assess stress, alertness, and various workload levels and is effective at assessing all* 4 HIP *bottleneck effects* • Implemented in both a military fast-jet simulation and a command and control application environment • Planned for implementation efforts in support of Boeing's AugCog program in their command and control closed-loop application domain	Appropriate for stationary users in stationary or mobile application domains; not yet tested with mobile users

**Note*: The above list is by no means all-inclusive. For a thorough review of available sensors, see NATO report on Operator Functional State Assessment (Wilson & Schlegel, 2004). Further, many of these tools may be used individually or combined in various ways to assess multiple HIP bottlenecks, as well as other HCI task factors (e.g., context, environmental stress effects, etc.).

in both the back and seat. Data from these sensors are used to compute changes in pressure over time, and the medial-lateral shifts of position in the seat. To record head position and changes, the head tracking component involves using a head-mounted wireless transmitter and four detectors, positioned on the corners of a computer monitor placed in front of the user. Posture shift data, head position, and head velocity, each output every 250 milliseconds, were used by LM-ATL to gauge levels of attention (e.g., engagement). Although a few experimental trials indicated ability to correlate with cognitive workload, LM-ATL determined more research would be needed to assess whether the body position/tracking system is reliably effective in predicting cognitive workload in command and control type tasks.

As mentioned later in the section titled "EEG," DaimlerChrysler has found success with this device by integrating it with their EEG- and contextual-based sensor suite. They based classification of workload conditions on the basis of the driver's behavioral data from the University of Pittsburgh system. Head tracking or monitoring of the driver's seat posture, for example, allowed DaimlerChrysler to detect preparatory movements likely to be followed by certain driving maneuvers. Combining this technique with environmental monitoring, they also found it possible to recognize situations indicative of certain (upcoming) cognitive states and thus develop proactive strategies to miminize cognitive loading effects.

Stress Gauge

The Institute of Human and Machine Cognition (IHMC) in conjunction with the Honeywell DARPA IWIIUS industry team (Kobus et al., 2005) has developed a composite Stress Gauge (Kass, Doyle, Raj, Andrasik, & Higgins, 2003; Raj, Perry, Abraham, & Rupert, 2003), which uses a weighted average of the following three sensor inputs to detect the participant's response to changes in cognitive load: Video Pupillometry (VOG), High Frequency Electrocardiogram (HFQRS ECG), and Electrodermal Response (EDR). This gauge has been used to track the autonomic response to time pressure in a high workload environment and to detect cognitive stress related to managing multiple competing tasks on a moment-to-moment basis. Thus, it is appropriate for assessing HIP bottlenecks related to attention (e.g., divided attention effects).

Arousal Meter Gauge

Clemson University in conjunction with the Honeywell and Boeing DARPA IWIIUS industry team efforts (Kobus et al., 2005) has developed an Arousal Meter (Hoover & Muth, 2003), which derives autonomic arousal from the cardiac interbeat interval (IBI) derived from the ECG at one millisecond accuracy. The gauge has three levels (low, medium, high), which increase or decrease with respective increases or decreases in autonomic arousal. The Arousal Gauge has been used to track decrements in performance due to low arousal states and is thus appropriate for assessing HIP bottlenecks related to attention (e.g., divided attention and vigilance effects).

eXecutive Load Index Gauge

In conjunction with the Honeywell industry team efforts, Human Bionics has developed the eXecutive Load Index (XLI) gauge (DuRousseau, 2004a, 2004b) (see Fig. 62.1), which utilizes EEG signals to allow measurement of patterns in tightly-coupled cortical networks tied to an individual's allocation of attentional resources as a user's cognitive state changes in response to conditional task load (Kobus et al., 2005). The index was designed to measure real-time changes in cognitive load related to the processing of messages and has been previously validated regarding ability to discern trial difficulty in a continuous performance, high-order cognitive task battery. The XLI gauge has been used to track the active inhibition of competing tasks, and is therefore appropriate for assessing divided at-

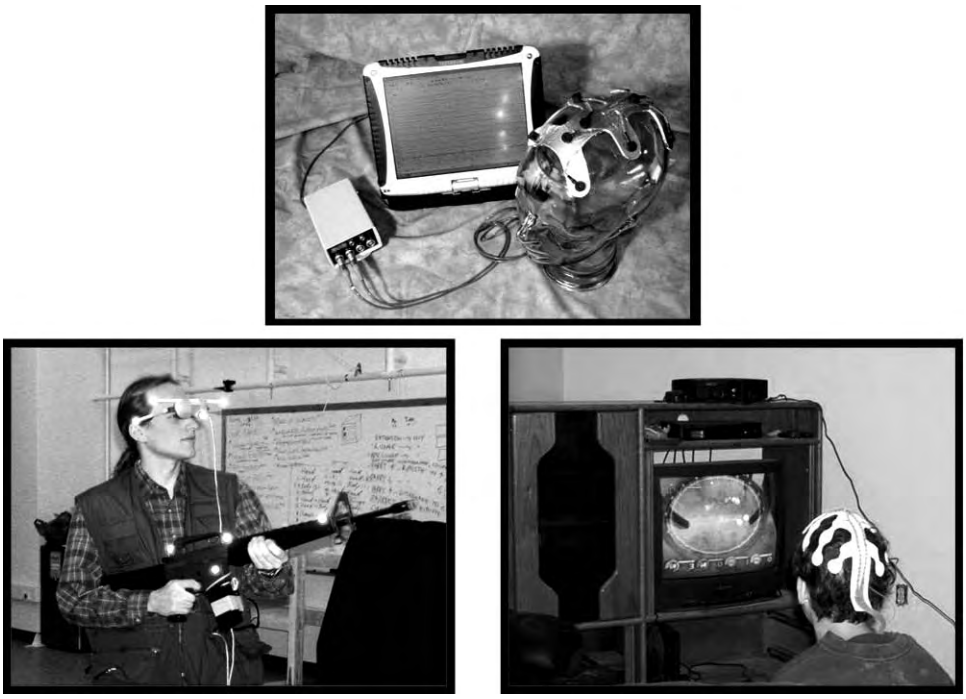


FIGURE 62.1. Examples of Human Bionics' XLI system.

tention effects on users. This gauge has also been effectively integrated into a general workload assessment tool.

P300 Novelty Detector Gauge

Columbia University and the City College of New York (CCNY) have created the P300-novelty detector gauge (Sajda, Gerson, & Parra, 2003) in support of Honeywell's DARPA IWIIUS program efforts (Kobus et al., 2005). This gauge spatially integrates EEG auditory signals from sensors distributed across the scalp, learning a high dimensional hyperplane for discriminating between task-relevant (e.g., incoming message auditory alert) and task-irrelevant responses. The EEG auditory P300 reflects a central nervous system response to behaviorally relevant infrequent sounds. Previous literature (Wickens, Heffley, Kramer, & Donchin, 1980) suggests that the P300 amplitude, in response to a task-relevant infrequent auditory stimulus, is modulated by attentional resources (e.g., if the participant is very focused on a primary task, the secondary auditory stimulus will be missed and the corresponding P300 diminished). User state assessment is based on the assumption that the presence of a strong evoked response indicates a user has sufficient attentional resources to process an incoming message. The gauge includes frontal and parietal electrodes and tracks the attentional resources used to attend to novel stimuli and is also an indirect measure of the response capacity to competing tasks and a user's narrowing attention (e.g., divided attention effects).

Engagement Index Gauge

The Engagement Index Gauge is a ratio of EEG power bands (beta/(alpha + theta)) and was developed with a combined effort from NASA, the City College of New York (CCNY) and Honeywell (Kobus et al., 2005). The engagement index is a measurement of how cognitively engaged a person is in a task (e.g., their level of alertness) and is described in more detail by Freeman, Mikulka, Prinzel, and Scerbo (1999). Adaptive systems have used this index for controlling the automation switching between manual and automatic modes (Mikulka, Scerbo, & Freeman, 2002; Pope, Bogart, & Bartolome, 1995). Prinzel, Hadley, Freeman, and Mikulka (1999) reported that adaptive task allocation may be best reserved for the endpoints of the task engagement continuum. Honeywell, therefore, implemented two levels of engagement (low, high) to reflect the selection and focus on some aspect at the expense of the other competing demands (e.g., focused attention effects). For instance, high levels of engagement reflect selection and attentional focus, whereas low levels of engagement indicate that the participant is not actively engaged with some aspect of the environment during the respectively assessed point in time. The Engagement Index Gauge has been demonstrated by Honeywell to be an effective tool for assessing both sustained (e.g., vigilance) and divided attention bottleneck effects on HIP and is particularly sensitive to low workload conditions.

New Workload Assessment Monitor (NuWAM; Combined EEG, ECG and EOG Sensors)

In conjunction with the Boeing team's DARPA IWIIUS program efforts, the Warfighter Interface Division of the U.S. Air Force Research Laboratories (AFRL) evaluated AugCog sensors (e.g., EEG, ECG, and EOG) and performed preliminary integration at the AFRL/HECP facilities at Wright Patterson AFB to form the NuWAM sensor suite. NuWAM is based on an artificial neural net (ANN) cognitive state classifier of a 20-channel EEG system along with the ECG and EOG sensors. Although not as effective at assessing specific HIP bottlenecks, the AFRL NuWAM gauge can determine general workload levels for specific types of tasks that have been categorized during cognitive task analyses.

Dry Electrodes

Various DARPA IWIIUS program team members worked with Quantum Applied Science and Research, Inc. (QUASAR) regarding their development of advanced dry electrode sensors for EEG, ECG, and EOG use. Boeing found QUASAR's ECG cloth chest strap worked very well as a fieldable ECG sensor system integrated with NuWAM. They also found the EOG eyeglass frame device to be non-invasive and very useful for holding the electrodes necessary for assessing vertical and horizontal eye movements.

Functional Near Infrared (fNIR)

Honeywell, Boeing, and LM-ATL industry teams have all worked with some form of an fNIR device (e.g., from NovaSol [now Archinoetics] and/or Drexel University), which measures blood oxygenation and volume changes in the brain relative to where the optical sensors are placed on the head (Kobus et al., 2005). Functional near infrared has been shown to be an effective tool for diagnosing cognitive activity associated with spatial and verbal working memory, given its often known right and left hemispheric separability, respectively (Smith & Jonides, 1998). Archinoetic's fNIR sensors, for example (see Fig. 62. 2), (a) are composed of a central emitter with eight detachable detectors, (b) are designed to be configurable in shape, and (c) use urethane fingers to penetrate through hair so that no preparation of the participant is required before using the system (and it may easily be applied anywhere on a user's head without adhesives). The emitter uses a multiwavelength LED and the detectors receive the fNIR signal with an amplified, highly sensitive photodiode. The detectors plug into the sensor with ribbon cable and are angularly mounted to the emitter with Velcro. The curvature formed by the angular mounting conforms to the shape of the scalp for maximum skin contact to minimize noise caused by light leakage.

Boeing's fNIR Approach

Boeing found that using the Archinoetics "whole head" fNIR sensors with statistical process control (SPC) cognitive state



FIGURE 62.2. Archinoetics (formerly NovaSol) fNIR sensor.

classifier algorithms proved very successful at detecting tasks categorized as primarily verbal (e.g., tasks with words, language, or logical operations; Sanders & McCormick, 1993; Kobus et al., 2005). Their fNIR/SPC approach gave qualitatively different results than their EEG/ANN approach previously discussed. A main difference observed between these approaches is that the fNIR/SPC approach acts more like a "trigger" and the EEG/ANN approach acts more like a gauge. They noted how a cognitive trigger differs from a cognitive gauge in that a trigger responds quickly to the onset of a task, but does not stay high for the duration of the task, which is reflective of HIP tendencies to respond to changes in the environment rather than remain at a steady state. Boeing found their fNIR/SPC approach to be most effective for assessing workload effects during episodic tasks, but it also reliably responded to other HIP bottleneck gauges. Although Boeing found their fNIR/SPC system to be a fast-acting cognitive trigger, they noted that this approach has a disadvantage in that it currently uses one sensor per SPC channel and thus lacks the redundancy of their EEG/ANN system. Boeing will continue to refine both approaches and investigate the potential for combining them to provide a more robust and sensitive gauge of cognitive state and changes in real time.

LM-ATL's *f*NIR A*pproach*

LM-ATL used an fNIR device provided by Drexel University, which is a Fast fNIR device that is mounted to a user's forehead in order to track changes in cognitive activity in the frontal lobe region of the brain (Kobus et al., 2005). The Fast fNIR device contains four light emitters surrounded by 10 detectors to measure the amount of reflected infrared light and the angle at which it is reflected. Thus, the amount of cognitive activity in the regions of the cortex below the device can be tracked and assessed. LM-ATL found the device to be useful in gauging levels of attention, verbal working memory, and overall workload.

In future AugCog efforts, Honeywell plans to explore fNIR capabilities for mobile users (dismounted soldiers) since early studies from Drexel and LM-ATL also indicate promise for being able to filter the data to remove motion artifacts (Dorneich et al., 2007; Kobus et al., 2005). Based on Boeing's results with NovaSol's fNIR device, both Honeywell and LM-ATL (for stationary command and control type tasks) may further explore its utility in their real work applications, assuming usability and ergonomic fit issues from the 'whole head' device can be minimized.

Pupillometry

The LM-ATAL and Boeing teams used EyeTracking, Inc.'s (ETI) Index of Cognitive Activity (ICA) system (see Fig. 62.3 and Fig. 62.4), which is based on binocular eye tracking sensors and



FIGURE 62.3. ETI sensor system setup. (Note: the Eyelink II is manufactured by SR Research Global, Inc.)

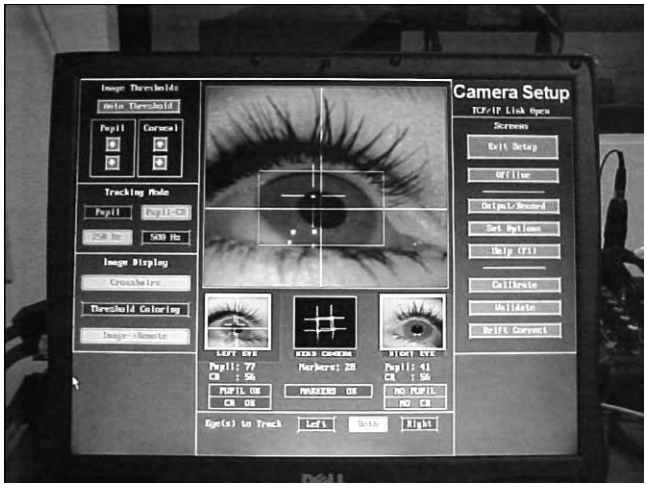


FIGURE 62.4. ETI eye tracking system. (Note: the Eyelink II is manufactured by SR Research Global, Inc.)

measurement of pupil dilation (Kobus et al., 2005) ETI's sensor configuration utilizes a lightweight headset with three small cameras in order to record exactly where a user is looking. The user's point of gaze is superimposed on a video recording of the displays. While users interact with a computer system, their eye movements and pupil dilation measures are monitored and recorded by multiple miniature high speed cameras (samples at 250 Hz). The eye patterns can also be plotted on a bitmap in order to look at a viewing history spanning several seconds or longer.

Although not a direct measure of cognitive activity, ETI's sensor technology applies proprietary and patented techniques to determine the level of cognitive effort, the amount of sustained attention, and the affective response of a workstation operator as based on their ICA, which is ETI's patented metric for estimating cognitive activity from changes in pupil dilation (e.g., abrupt changes in pupil diameter indicate an operator's current levels of mental effort, while the point of gaze metric highlights specific elements causing difficulty).

For Boeing's efforts, ETI also provided a data capture system that could integrate and display in real time their multisensor data (e.g., EEG, fNIR and eye-tracking systems). This allowed them to produce high fidelity estimates of cognitive state change, even though a direct measure on an HIP bottleneck is not presently possible with ETI's system.

LM-ATL found that it may be possible to determine different levels of utilization of left vs. right brain hemisphere regions by comparing pupillometry data from the different eyes. Such data could be used as an indicator as to whether a user was using more verbal (left hemisphere) or spatial (right hemisphere) processing resources. Consequently, ETI's device also shows promise for use as a gauge of the working memory HIP bottleneck.

EEG

Out of all brain imaging methods currently available, EEG is well documented and is proving to be a robust neuro-based tool for real-time system application, given its temporal resolution and potential spatial resolution when combined with other techniques. All the DARPA IWIIUS industry teams used some form of EEG device for their program efforts (Kobus et al., 2005). As discussed previously, Boeing and Honeywell have used EEG sensors as part of integrated sensor suites to gauge various states of cognitive activity and HIP bottlenecks. Efforts involving LM-ATL's and Daimler Chrysler's EEG-based approaches are also showing very promising results for assessing cognitive state in real time.

LM-ATL's EEG-Based Technology

LM-ATL used Advanced Brain Monitoring's (ABM) EEG device, which is a portable EEG cap, wireless transmitter, and B-Alert software used to collect continuous EEG signals (see Fig. 62.5). Various models of the device are available from ABM, but LM-ATL has implemented versions using three, six, and nine EEG electrodes for different experiments (Kobus et al., 2005). The cap contains a wireless transmission unit which sends measures of electrical activity (in microvolts, μV) collected by each electrode to a laptop computer once every millisecond. LM-ATL's use of ABM's EEG device has contributed to the development of gauges capable of estimating various types of cognitive states, including vigilance/arousal, workload, engagement, distraction/drowsiness, and verbal working memory. LM-ATL also found that using ABM's continuous EEG device to estimate cognitive state of an individual requires only 15 minutes of baseline data to be collected for that individual, indicating feasibility for various fielded experiments.

LM-ATL has also implemented an ERP EEG system, which was developed by ElectroGeodesics, Inc (EGI). Instead of a cap, as provided with ABM's system, EGI's system involves users wearing a net containing either 128 or 256 embedded electrodes. The EEG-net covers the entire hairline, forehead, and parts of each cheek, and its electrodes capture electrical activity, in Microvolts (μV), in the cortex below each electrode. Comparing measures for different pairs of electrodes before and after tasks of interest provided LM-ATL with estimates of which



FIGURE 62.5. Examples of ABM's EEG cap in use.

brain regions were engaged and to what degree as users performed various tasks. These data were then processed to help establish verbal working memory, spatial working memory, and cognitive workload gauges. LM-ATL found that using EGI's ERP-EEG based system allowed accurate localization of cognitive activity in an individual with approximately five minutes of baseline data. While its greater intrusiveness and incompatibility with other LM-ATL sensors as compared to ABM's device may limit its application in certain task environments, the rapid baselining and measurement sensitivity provided by EGI's device makes it feasible for research applications and possibly real-world clinical applications.

DaimlerChrysler's EEG-*Based Technology*

In conjunction with FIRST of Berlin, Germany, Daimler-Chrysler has developed an EEG-based integrated sensor suite system to estimate a driver's cognitive state in a Mercedes-Benz vehicle (Kobus et al., 2005). The EEG signals collected from the driver are also supported by EOG and electromyography (EMG) recordings to facilitate artifact recognition. The EEG is sampled at 500 Hz from 29 scalp electrodes arranged according to the international 10–20 system. For DaimlerChrysler's AugCog program efforts, the driver's seat posture and head movements are also captured by a dedicated seat mat and a head tracker device as provided by the University of Pittsburgh, while another group of sensors are devoted to capturing task context and estimating its influence on the driver's cognitive state in a system devised by Sandia National Laboratory (Kobus et al., 2005). Values from the EEG signals and the other three vehicle-sensor groups (which are integrated via a Controller Area Network [CAN] data bus), are classified via algorithms whose outputs represent interpretations of the driver's current workload (high or low) or of the context's impact on it, respectively. DaimlerChrysler has found that their combination of both EEG-based neurophysiological and contextual-based physiological sensors provides a good measure of cognitive state correlates (e.g., overall workload, attention and sensory HIP bottleneck effects), along with the ability to monitor and adapt to a wide variety of environmental factors (via the vehicles built-in sensor system) in dynamic, critical driving situations (Kobus et al., 2005). They have also noted that further refinement of these gauges for closed-loop AugCog system implementation would be desirable, including ability to have: a higher temporal resolution of the EEG-based classifiers, without losing classification accuracy, and; a finer gradation of workload levels (at least low, medium and high).

Event Related Optical Signal (EROS)

Boeing investigated the potential for using a new fast optical imaging tool based on an EROS system (e.g., reflected light from the cortical level is measured and reconstructed into a map of brain activity), which is being researched and developed at the University of Illinois (Gratton & Fabiani, 2001; Kobus et al., 2005). Initial investigations involved using EROS to aid in identifying brain signatures of preparation that arise when a cue dictates a user's covert attentional shift to the left or the right visual field when a correspondingly mapped precue is used to signal a user to shift their attention (but not their eyes) to the left or the right side of fixation. The University of Illinois has found a basic switch effect in the right parietal cortex whose region showed greater activation when the cue signaled a shift of attention to the opposite hemifield from that attended on the previous trial. The time course of this activity showed both an early response (115 millisecond post cue) and a later one (354 ms) that was similar in timing to values previously seen with other neurotechnologies.

Further, and consistent with typical contralateral mapping of hemifields to cortex, the right hemisphere showed activation while the left hemisphere showed inhibition when the precue signaled a leftward shift of attention and vice versa. These lateralization effects occurred in a region commonly referred to as the frontal eye field and began quite early (56 milliseconds) and peaked at approximately 150 milliseconds. In essence, Boeing found the EROS measure to be sensitive to both general and direction-specific attentional preparation and thus may be a good gauge of sensory input, attention and executive function HIP bottlenecks during cognitive state assessment in future applications. Table 62.1 provides a summary of the advantages of this and the aforementioned techniques that have been effectively implemented by AugCog researchers and developers to date.

Cognitive Monitor (CogMon)

QinetiQ has designed the CogMon sensor suite system (Dickson, 2005a) whose primary function is to provide estimations of a pilot's cognitive-affective status in near-real time while interacting in a fast-jet cockpit simulation environment. Pilot state inferences are derived from four main sensor sources (e.g., behavioral measures from interactions with cockpit controls, EEG-based physiological measures, subjective measures, and contextual information), which are used to assess the pilot's objective and subjective cognitive state in terms of arousal and workload levels (Pleydell-Pearce, Dickson, & Whitecross, 2000). Cognitive state estimations are encapsulated within high-level state descriptors such as levels of stress, alertness, and workload, and are then provided to a Tasking Interface Manager (TIM) to aid in directing levels of automation or information presentation within the cockpit. CogMon can be configured to enable both online and post-hoc operation.

APPLICATION OF NEUROTECHNOLOGIES

While other application areas may emerge in the near future, the three most appropriate application areas that may presently benefit from effective implementation of current neurotechnologies include: (a) Operational closed-loop systems, (b) Design of Systems, and (c) Training/Education Systems. This section reviews some of the S&T developments in these areas and implications emerging from the DARPA IWIUS program and various AugCog-related efforts.

Closed-Loop Systems

Leveraging physiological and neurophysiological technologies (such as those that are emerging from the DARPA IWIIUS program and maturing via real-world application, as discussed in the previous section), it is now possible to assess the ever-changing cognitive state of the user in real time and mitigate against human performance limitations caused by known HIP bottlenecks. Recently, it has been demonstrated that the loop can be closed between this assessment and the adaptation of the user interface in real time to accommodate the assessed bottleneck. This is known as a closed-loop AugCog system and is the core of the DARPA IWIIUS program vision and of the Augmented Cognition field in general. Essential elements of any closed-loop AugCog system (see Fig. 62.6) must therefore include

- Cognitive state assessment capabilities via physio- and neurophysiological tools as discussed in the previous section.
- Methods for classifying the cognitive state data from these tools to allow mitigation of HIP bottleneck effects via adaptive automation techniques (e.g., AugCog mitigation strategies).
- An integration architecture that synchronizes and optimizes all necessary components via a robust controller[2] that maintains system stability (e.g., controlling the "when" and "how" of the adaptive automation techniques).

As summarized next, all industry teams in the DARPA AugCog-based IWIIUS program have demonstrated (a) the possibility of reliably identifying different cognitive states on the basis of neuronal signals, (b) that this information can be used to drive mitigation measures in human-machine interaction under closed-loop conditions, and most importantly, (c) that a system making use of these neurophysiological-based technologies is able to significantly improve the cognitive performance of human operators in multiple task domains.

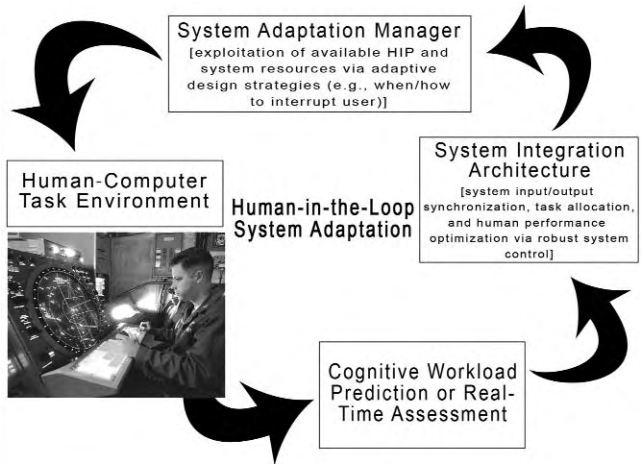


FIGURE 62.6. Essential elements of a closed-loop AugCog system.

Honeywell's AugCog Closed-Loop Efforts

As directed by DARPA, the Honeywell research team addressed *attention* as their primary HIP bottleneck in the context of a ground infantry soldier in the U.S. Army's Future Force Warrior (FFW) program (Kobus et al., 2005). The FFW will have more information available (via physical sensors, communications, etc.), and processing that information in a dynamic and lethal environment requires careful management of limited attentional resources. During their Concept Validation Experiment (CVE) they demonstrated a 380% performance improvement where attention resources were required, exceeding the 100% improvement goal set by DARPA. Honeywell was able to correctly classify attention state changes more than 98% of the time and in less than 300 milliseconds. Their success was due to their use of the following:

- Cognitive state assessment tools (e.g., Stress Gauge, Arousal Meter Gauge, XLI, P300 Novelty detector, Engagement Index, fNIR, and EEG);
- Data classification methods (e.g., spectral analysis techniques and linear regression on weighted means);
- An augmentation manager to drive mitigation strategies (e.g., task/information and modality management, task offloading, and task sharing), and;
- An agent-based integration architecture.

Honeywell's AugCog S&T closed-loop approach could provide improved human performance and enhanced survivability for most any command and control system operator (e.g., stationary command center, or individual/mobile dismounted soldiers), given the core capability is the availability of netted communications and mitigation strategies enabling sharing and real-time collaboration, all of which enhances the kind of situation awareness (SA) necessary to drive decisive actions in diverse and stressful operational environments.

DaimlerChrysler's AugCog Closed-Loop Efforts

DaimlerChrysler was directed by DARPA to address the *sensory input* HIP bottleneck in the context of a Light Armored Vehicle (LAV) operator (Kobus et al., 2005). A future LAV operator will have multiple cognitive tasks (e.g., communication, planning, command and control) to manage. Sensory input resources are required by these cognitive tasks as well as by the primary job of maneuvering the vehicle. DaimlerChrysler investigated the sensory bottleneck in the auditory and visual modalities, with multiple

[2]In a forthcoming AugCog International Conference Proceedings publication (Reeves, Young, Schmorrow, & Stanney, 2007), AugCog researchers and developers have defined a robust controller to be a technique that uses appropriate methods and applied theories to ensure real-time systematic control and stabilization of an AugCog system to significantly improve human-system performance.

mitigation strategies. Their experimental results under real-world driving conditions showed that the sensory bottlenecks could be improved by as much as 108% with an accuracy of up to 98%, depending on the modality being examined. The sensory bottleneck status could be detected in as little as 200 milliseconds, and mitigations could be invoked in as little as 0.2 seconds, depending on the mitigation being used. Their AugCog closed-loop successes were due to their use of the following:

- Cognitive state assessment tools (e.g., EEG-based working memory and visual/auditory sensory input gauges; a combination of context, driving behavior, and seat posture sensors).
- Data classification methods (e.g., Linear Discriminate Analysis [LDA] and ANN).
- An augmentation manager to drive mitigation strategies (e.g., task/information and modality management, task scheduling).
- An integration architecture based on both stand-alone gauge processors and a vehicle CAN data-bus technology.

DaimlerChrysler's AugCog S&T closed-loop approach could be used to facilitate the design of most any in-vehicle systems that need to be interoperable and adaptable in real time to an operator's changing cognitive abilities and limitations instead of the operator having to adapt to inefficiently designed vehicles. Potential human performance payoffs for vehicle operators would include (a) increased operator situational awareness, (b) improved asymmetric/intuitive decision making, and (c) improved attention management under stress.

LM-ATL's AugCog Closed-Loop Efforts

LM-ATL was directed by DARPA to address *working memory* as their primary HIP bottleneck within a Tactical Tomahawk Weapons Control System (TTWCS) simulation environment (Kobus et al., 2005). The essential TTWCS operator, the Tactical Strike Coordinator (TSC), must manage a number of missiles, targets and shipboard launch platforms, and dynamically reassign the missiles to targets as critical targets pop-up, missiles fail, and so forth. The operator must recall and recognize far more information than can be maintained in working memory in order to perform the task effectively. As part of their CVE efforts, LM-ATL demonstrated an improvement in working memory throughput by at least 500% by using an Intelligent Sequencing mitigation strategy to strategically present related information about specific missile-target pairings when the working memory bottleneck was saturated. Working memory status (high or low cognitive workload) was correctly identified in over 90% of the trials, with the sequencing mitigation being invoked in less than 500 milliseconds. LM-ATL's AugCog closed-loop successes were due to their use of the following:

- Cognitive state assessment tools (e.g., EEG-based working memory gauge, EDA/GSR- and EKG [ECG]-based based arousal and cognitive workload gauge, pupillometry, and fNIR).
- Data classification methods (e.g., ANN).
- PACE (Performance Augmentation through Cognitive Enhancement) processor to manage tasks and enable mitigation strategies (e.g., modality management and intelligent sequencing).
- An integration architecture based on both the rule-based PACE architecture and an ANN.

LM-ATL's AugCog closed-loop system was designed to allow rapid transition of PACE into commercial and DoD acquisition activities and to enable system operators to maintain a state of high alert without overtaxing (or undertaxing) their cognitive abilities. LM-ATL's AugCog S&T approach could provide improved human performance for most any command and control system operator.

Boeing's AugCog Closed-Loop Efforts

The Boeing team was assigned the *executive function* HIP bottleneck in the context of an Unmanned Air Vehicle (UAV) systems operator task. Executive function is analogous to the central processor in a computer, and relates to the need to process and rapidly assimilate many information elements for a single operator to manage multiple UAVs (Kobus et al., 2005). Boeing developed an array of mitigations that were triggered to address their primary cognitive bottleneck and others (e.g., attention, sensory input, and working memory). With regard to their targeted bottleneck, they empirically showed a 241% performance improvement in executive function-related tasks, with a classification accuracy of 92% in less than one second. The mitigations were also determined and triggered in less than one second. The Boeing team's AugCog closed-loop successes were due to their use of the following:

- Cognitive state assessment tools (e.g., body position/posture tracking, Arousal Meter Gauge, NuWAM combined EEG, ECG, EOG sensors, fNIR/SPC, EEG/ANN, and pupillometry).
- Data classification methods (e.g., SPC and ANN).
- An augmentation manager to drive mitigation strategies (e.g., task/information and modality management, task scheduling/sequencing, and map declutter).
- An integration architecture, where sensor/gauge data are collected via a networked database and used to trigger rule-based mitigation strategies).

Boeing's AugCog closed-loop S&T approach could be applicable to most any command and control system, not just the UAV AugCog prototype workstation, for improving operator performance. Their closed-loop AugCog system was designed for rapid transition to other domains (e.g., Air Traffic Management workstations, Airborne Early Warning and Control System [AQACS], and Airborne Early Warning and Control [AEW&C]).

QinetiQ's AugCog Closed-Loop Efforts

The latest team member added to the DARPA AugCog program was QinetiQ, whose closed-loop AugCog approach has been being implemented in two different contexts: (a) a military fast-mover jet simulation environment, and (b) a simulated UAV

command and control task environment, where their AugCog S&T approach was planned to be adapted to support the Boeing industry team's AugCog efforts involving their UAV command and control task domain (Dickson, 2005a; Schmorrow & Kruse, 2005). QinetiQ's system is referred to as the Cognitive Cockpit (COGPIT). It was developed in support of the vision of a cockpit that allows a plot to concentrate skills towards the relevant, critical mission event, at the appropriate time and to the appropriate level, by ensuring that the correct balance of skill and knowledge-based tasks are appropriately divided between and among the system and the operator (Dickson, 2005b; Taylor, Howells, & Watson, 2000). QinetiQ uses a variety of techniques to support the dynamic yet stable distribution of tasks between the system and operator (e.g., attentional management, adaptive interfaces, adaptive automation, decision support, task scheduling, and context-sensitive support of responses from interactions with cockpit controls). The success of QinetiQ's seamless cockpit prototype system, COGPIT, to be able to assess and accommodate a pilot's cognitive task load within challenging ground to air attack scenarios, is due to their advanced AugCog S&T approach including the use of the following (Dickson, 2005a, 2005b):

- Cognitive and context state assessment tools (e.g., CogMon).
- Proven and accurate data classification methods.
- A decision support system whose knowledge base is derived from structural interviews with subject matter experts (SMEs), tactical manuals, and rules of engagement.
- An adaptive interface manager (e.g., the Tasking Interface Manager [TIM]), which is based on an adaptive automation framework, the Pilot Authorization and Control of Tasks (PACT; Bonner, Taylor, Fletcher, & Miller, 2000).
- A flexible, modular agent-based integration architecture.

QinetiQ's approach represents a culminating effort that brings together all the tenets and the vision of the AugCog field into a single, fully integrated closed-loop system. Their approach could be used as a model for most any closed-loop application domain, as well as for other HCI design goals. The next section addresses how AugCog S&T emerging and maturing from QinetiQ's, as well as the other four main industry team's DARPA program efforts, may be used for enhancing the HCI design lifecycle and provide more diagnostic system evaluation capabilities.

Improving HCI System Design and Evaluation Capabilities

The ability to build systems truly tailored to a user's HIP needs and capabilities is now possible. Based on the review of AugCog technologies presented in the previous sections, it is evident that HCI practitioners now have available to them various physiological- and neurophysiological-based tools and techniques that may be used to better specify both the design and evaluation of human-computer systems. LM-ATL, for instance, as part of the Office of Naval Research's (ONR) Disruptive Technologies Opportunity Fund (DTOF) program, is currently exploring ways to leverage some of their AugCog technologies as tools for improving future TTWCS interface designs and establishing more diagnostic system evaluation (e.g., usability) methods for integration into their system design lifecycle.

Using cognitive state gauges, LM-ATL will be able to assess in real time the load on various HIP bottlenecks while users interact with prototyped simulations or existing systems—instead of having to predict potential overload issues and/or infer potential HIP workload issues based on resulting human performance effects, as in traditional HCI design and evaluation studies. The neurophysiological-enhanced studies may involve either: closed-loop systems, where potential alternate adaptive strategies are compared in terms of their HIP workload and subsequent human performance effects in real time, or; open systems, where an existing interface design's efficiency and effectiveness is assessed in real time in terms of HIP workload and subsequent human performance effects as a user interacts with it.

Most any of the neurophysiological tools and techniques implemented by all of the DARPA IWIIUS industry teams could be used and applied to the design and evaluation of most any human-computer system, resulting in more rapid, effective, and eventually less expensive HCI design and evaluation processes. Such benefits could also be achieved in training system application areas, as discussed next.

Training Systems and Operator Selection

Given the promising results realized thus far by the AugCog industry teams, the ability to better understand and tap the neurophysiological correlates of cognitive state holds great promise for benefiting human performance, training, and education (HPTE) efforts through ONR's HPTE Program—the ability to tailor training approaches to the needs and capabilities of individuals within any given group and dynamically adapt approaches, potentially in real time, during actual training sessions. This section reviews particular ONR Cognitive Performance Enhancement (CPE) initiatives and future directions in potential operator selection methods that may be directly impacted by AugCog S&T that has emerged from the DARPA program and the Augmented Cognition field in general.

CPE I*nitiative*[3]

The CPE efforts (as part of ONR's HPTE program) focus on developing training systems and novel training principles that accelerate and enhance the training progression (e.g., novice to expert) of both individual Marines and teams of Marines working in decision making environments. The main objective is to develop tools to support real-time cognitive and behavioral

[3]Adapted from 2005 ONR HPTE Program Review brief by Schmorrow and Stripling and various performer briefs, CPE program briefs, and the following preliminary AMSUS abstract by Stripling, Cohn, & Kruse: Cognitive Performance Enhancement: Leveraging Neuro-cognitive Principles to Accelerate the Transition from Novice to Expert.

assessment and improvement of individuals and teams during operations and training. Implicit is the need to develop systems that will enable the ability to more effectively screen individuals based on their decision-making ability or their capability for mastery of decision making skills. Addressing such a need could substantially reduce training time and costs currently invested and necessary to get military personnel to deploy-ready levels.

Assessing training progression. There are several potential sources of objective metrics that could be used to support adaptive training and assessment of deploy-readiness of military personnel. A promising yet largely untapped source of such metrics are neurophysiological data that are processed in real time to identify cognitive states and stages of development important to the training process. Years of basic research have identified numerous markers of attention, engagement, and cognitive load that systematically change as task proficiency develops. Various noninvasive brain monitoring technologies have been successfully applied to the challenge of documenting the transformation from novice to expert in a variety of domains, including: identifying indices of skill level in basic laboratory tasks and marksmanship (Ciesielski & French, 1989; Deeny, Hillman, Janelle, & Hatfield, 2003; Kerick, Douglass, & Hatfield, 2004); identifying indices of skill acquisition in computer games (Smith, McEvoy & Gevins, 1999), and; detection of the progression towards automaticity of syntactic processing (Gunter & Friederici, 1999). Recent studies have also documented brain activity patterns that are indicative of a shift in cognitive processing from focused conscious effort to "automatic" processing (Floyer-Lea & Matthews, 2004; Peres et al., 2000). This shift to automatic processing may occur after task performance has leveled off and is significant because it marks the point at which minimal cognitive effort is required to perform the task well. This transition may serve as an indication of the point at which the trainee is ready to learn additional components of the task or be taught higher level strategies within the particular domain.

The identification of neurophysiological measures that correlate with task and skill mastery is of value for three practical reasons:

1. As described above, it could serve as the basis for real-time closed-loop training systems that adapt to the individual's current state of task proficiency and automaticity; this application would have the most profound impact on the training capabilities, but is not ready for widespread distribution.
2. Prior to widespread distribution for closed-loop training, AugCog-based technologies could be used to evaluate alternative training strategies and systems to identify those that provide the most effective learning environment. This application is potentially feasible in the near future, but would require more upfront research to confirm the neurophysiological patterns associated with task mastery are applicable to more complex task environments.
3. During the course of this type of research, novel training strategies may emerge that are validated by laboratory observations of accelerated task mastery and automaticity, but which do not rely on real-time monitoring of brain activity to enact. Such observations would potentially be immediately transferable to instructional environments.

Using AugCog technologies, such as sensors (e.g., EEG, ECG, fNIR, pupillometry) and gauges (e.g., EEG-, fNIR, or EROS-based cognitive workload gauge) employed by the AugCog industry teams to assess how novices become experts, or to identify when a soldier may be deploy-ready, could have significant impact for benefiting most any HPTE, including:

- Enhanced instructor understanding of a trainee's capabilities and limitations
  - More suitable operator assignments to match skill levels
  - Ability to predict fast learners early during training
  - More timely delivery of information provided to trainee
  - Increased learning capabilities (e.g., via an increase of an individual's cognitive processing capabilities during training)
- More expertly trained and mentally prepared operators
- Enhanced training acquisition
- More effective training per unit time (reduce required financial investment)
- Enhanced development of effective individual and team performance (possibly in real time)

Assessing individual potential (e.g., AugCog-enabled screening tools). Similar AugCog S&T approaches used for assessment of training progression could also be used to determine an individual's cognitive potential in general or their potential to reach a certain skill level in a particular task domain. For instance, the military has been using screening tools for decades to select for fighter pilots before thousands of dollars and man hours are wasted in training someone who may never be cognitively equipped to attain the necessary skill level required for such a domain. AugCog enabled screening tools could be used for similar purposes to improve the diagnostic capabilities of existing screening strategies.

Assessment of capabilities via an extensive neurophysiological and neuropsychological/psychometric components skills battery would be required in such an effort. An example of such an approach could be to determine the neural correlates of the Knowledge, Skills and Attitudes [and/or Abilities] (KSAs) required for a particular task domain and training level. Individuals with the greatest human performance results respective to the required KSAs for the particular task domain could be used as the expert model with which to compare other individual's neurophysiological indicators and patterns. This approach would also need to leverage the novice-to-expert training progression assessment techniques discussed above to establish the KSA neuro-level correlates along the entire training progression continuum (see Fig. 62.7) for necessary comprehensive comparative evaluations. Individuals could be assessed initially to determine a priori how suited they might or might not be for a particular task domain (e.g., aptitude assessment). They could later be assessed to determine how their training is progressing at both the human performance and neuro levels. The KSA approach is only a recommendation, as other neuro-level correlates of performance and aptitude may also be used. Specific benefits of any neuro-level screening/selection approach could include:

- Identifying which individuals may be most apt to reach deploy-ready or expert levels for a given task domain and/or who may never have the "aptitude" to get to required levels.

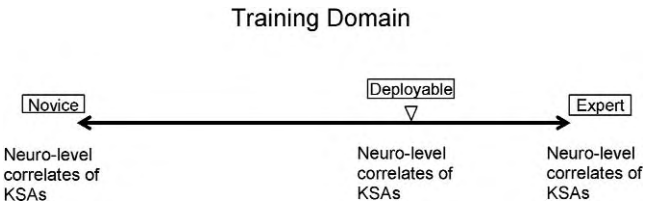


FIGURE 62.7. Conceptual model for leveraging neural correlates of training levels.

- Identifying operators with the greatest capacities for information processing and decision making, and then assigning them to critical information operations.
- Identifying users who could be trained to deployable- or expert-level more quickly than others.
- Reassessing trained individuals to see when retraining may be necessary (e.g., to reacquire necessary KSA levels).

The next section reviews the lessons learned to date from the DARPA IWIIUS program industry teams in pursuit of their HCI goal to build closed-loop operational AugCog systems. Such issues may also directly impact other the other main HCI application areas (system design and evaluation and training system applications).

LESSONS LEARNED AND FUTURE DIRECTIONS

AugCog technologies are at various levels of maturity. Thus, for the promises of the AugCog S&T to be fully realized, HCI practitioners need to consider the state of current technologies and the lessons learned from the DARPA IWIIUS program researchers and developers to date. To facilitate an assessment of the state of the field of AugCog to date, workshops involving principal AugCog scientists and engineers were organized at the March 2005 DARPA AugCog Program Review and at the Augmented Cognition International Conference in Las Vegas, Nevada in July of 2005. The summary papers resulting from the first workshop addressed AugCog S&T maturity levels and lessons learned for various areas and are summarized as follows in conjunction with the industry teams' DARPA IWIIUS program reports (Kobus et al., 2005).

The need to understand the user. A cognitive task analysis should be conducted in order to fully understand the decision-making requirements of the operator. Having a clear understanding of who the user is, as well as thoroughly understanding what they do, is essential for successfully constructing future AugCog technologies.

Integrate early, integrate often. The earlier and more thoroughly system integration (e.g., of the various sensors, computer hardware and software) was considered, the fewer the setbacks and more effective the mitigations. Slight modifications in sensors should not be treated superficially and require systematic assessment throughout development.

Development strategy. A project execution model that effectively balances a mix of execution strategies, where a distinction is made between developmental experimentation and formal empirical testing, seems best. AugCog Industry teams that were highly successful conducted step-wise assessments identifying the contributions to each individual modification of sensor/gauge implementation.

One sensor/gauge does not fit all. Some of the gauges that were thought to be the most robust during Phase 1 of the DARPA IWIIUS program were found to be less effective and/or useful when they were applied to the various operational requirements in Phases 2 and 3. Conversely, other gauges that were considered "marginal" during the Phase 1 technical integration experiment (TIE) have provided very effective results and operational utility. Further, gauges that seemed to work well for one of the teams in their application sometimes worked poorly for other teams within their application. In other words, the selection of specific gauges for use within the operational environment is highly context (task) dependent.

Cognitive bottlenecks may be task specific. The notion of isolated cognitive bottlenecks has significant appeal from a theoretical standpoint; however, it appears that when applying these concepts to real-world tasks, they are often overlapping or interacting in nature. It might be more useful to define bottlenecks and gauges in terms of the tasks that are to be mitigated vice attempting to address conceptual bottlenecks that are not adequately defined in terms of the operational task. It should be noted that regardless of the operational definition of a specific bottleneck, the net result of the mitigation strategies employed by AugCog teams improved overall task performance.

Artifact detection and correction. A main concern of artifact detection that is an urgent concern for the future advancement of AugCog S&T is the ability to decipher between physiological changes related to cognitive activity vice the physical requirements of the task. Such an issue may be even more of a concern for applications involving mobile users. Honeywell, for instance, noted that in applications involving mobile users, sensors can become loose and provide intermittent signals (and the data are directly affected by the motion—simple actions such as blinks can have large EEG signal effects). They need to be able to reduce the effect of the motion artifacts in order to continue to provide adequate estimations of cognitive state.

Perceived stability/trust. An issue with stability and predictability of AugCog mitigation strategies concerns the use of a combination of physiologically based gauges in conjunction with context-based sensors. If an operator does not understand the logic (e.g., the rationale for triggering the onset/offset of the mitigation), there is a significant potential for disruption and degraded performance. Another issue relates to the impact of degraded sensors. If users are not aware of the degraded functionality of the system, their expectations of augmentation would not be met, and the system may be perceived as unstable. Before such a system is deployed, training strategies will need to be developed to allow users to learn how to operate in the face of degraded augmentation.

Timing of mitigation strategies. Many of the changes in perceived cognitive load may be momentary. The physiological measures being used as gauges by the industry teams may detect either a transient or sustained change in cognitive activity. It may be unrealistic, or at least have little practical application, to have a system that is sensitive enough to detect and implement mitigations in terms of seconds. It is clear that in the operational environment, momentary changes in one or more of the cognitive bottlenecks areas may occur rapidly (<1 minute). More research needs to be conducted to investigate optimal timing for implementing various mitigation strategies.

Mitigation strategy principles. Mitigation strategies to date have been designed to address a specific HIP bottleneck to achieve specific performance objectives (e.g., avoid overload, increase throughput, offload, adapt to individual differences). All teams found that each mitigation strategy is beneficial in some situations while not under other circumstances, and thus must be regulated by human physiological and neuro-physiological indicators such that it does not undermine but rather enhances human performance. Consequently, the following mitigation strategy principles may be useful for future closed-loop AugCog development efforts:

- Pursue only those mitigation strategies that affirm the goals of AugCog (e.g., those that extend, by an order of magnitude or more, the information management capacity of the human-computer integral).
- Seek only to implement mitigation strategies based on objective and scientifically valid human performance assessment.
- Pursue only those mitigation strategies that are operationally feasible (either now or in the foreseeable future).
- Identify how/when individual differences will affect appropriate mitigation strategy choices and determine how to manage these differences such that human performance is enhanced for all.
- Enhance the effectiveness of mitigation strategy implementation by identifying which strategies may be user, context, or domain dependent and which are generalizable across these dimensions.
- Leverage the Arts to develop truly innovative mitigation strategies, particularly as AugCog S&T is applied to new display devices (e.g., PDAs, augmented reality systems, etc.).

Hardware integration. Results in Phase 2 by the four development teams indicated that both the underlying phenomena that the gauges are based on, as well as the artifacts that need to be removed for reliable gauge performance, may require fielded systems to be calibrated before each use. Future efforts will need to consider how to streamline the calibrations process and make it feasible under field conditions. Further, all teams noted the need to address making AugCog technologies more easily donned and readily calibrated. Significant consideration must also be given to potential interference sources in the field (particularly when wireless devices are used), and potential design solutions that could avoid such issues in the first place. Integrated sensors designed to be used in the field from the outset are more likely to adequately address this issue than technologies adapted from laboratory or medical applications. System integration architectures will need to be refined and streamlined in order to improve new sensor integration and to reduce the processing demands on any emerging sensor integration requirements.

Context gauges. Most of the AugCog S&T developments within the IWIIUS program focused on cognitive state gauges. It is becoming evident that these are necessary, but not sufficient, for effective AugCog systems. Each of the teams identified the need for "context gauges" (e.g., algorithms and sensors that would allow the augmentation system to have inputs regarding what is happening in the environment in which the user is interacting). Such inputs could be critical in determining which of several mitigations might be applied, as well as when it is appropriate to turn off the mitigation.

Individual differences. The teams reported significant issues with both interindividual and intraindividual differences, particularly for the EEG-derived gauges. Another aspect of the underlying EEG phenomena that is poorly understood is the effect of extensive experience with the tasks on the utility of EEG-derived gauges. It appears that successful utilization of EEG data in an AugCog system may require that the filters used in separating signals from artifacts will need to be tailored to every user and even be dynamically adaptable during the course of use of the AugCog system.

Proactive vice reactive augmented cognition. AugCog closed-loop systems developed to date are largely reactive in that the user must first become overloaded (or at least close to overloaded) before the system will invoke a mitigation strategy. Although it is important to detect when such levels of activity occur, it would be of great operational importance to develop systems that are also predictive (proactive) rather than simply reactive.

Metrics. All teams had difficulty in determining the appropriate level of analysis for assessing the impact of their AugCog closed-loop system. The difficulty stemmed from whether trying to assess the effectiveness of detecting changes in the cognitive bottleneck gauges, the effectiveness of the mitigations, or the effectiveness of the overall human-machine system in performing a task. Thus, the definition and use of appropriate metrics for AugCog-enabled systems will continue to be an important issue.

Prior to and during the second workshop held at the AugCog International conference, key scientists and engineers provided an initial survey-based assessment of the field of AugCog. Overviews of the survey and workshop results are presented in upcoming conference proceedings publications (Reeves, Schmorrow, & Stanney, 2007; Reeves, Young, Schmorrow, & Stanney, 2007). The survey and workshop focused on achieving consensus on the definition of AugCog and the various components that comprise its R&D base, as well identifying impediments to the field's current and future R&D. Some of the survey's and workshop's key results are summarized next.

Cognitive State Sensors

There were many varied issues reported regarding important challenges to the effective *design* of cognitive state sensors, all of which culminated in the main theme that the basic sensor technology on which AugCog technology is based still requires basic and applied research in many areas, including:

- Design of the sensors for ease of use, calibration, appropriate resolution/sensitivity, noise cancellation, and less invasiveness.
- Better understanding of neuro, psych, and cognitive theories that should be driving sensor placement, data analysis, and subsequent "cognitive load" and/or "cognitive state" gauge derivation.
- Determining appropriate experimental techniques in applied task settings to assess effectiveness of sensors to accommodate both general use settings and individual differences across task domains.

Many of these same issues were reiterated when discussing important challenges to the effective *use* of cognitive state sensors, with wearability, accounting and accommodating for day-to-day and individual user variability, and the development of appropriate signal processing algorithms and their interpretations being additional points of concern. When discussing important challenges to the effective *adoption* of cognitive state sensors, many of the same issues were again highlighted, with user acceptance (e.g., ease of use, comfort, safety, trust, control, etc.) and fieldability (with value-added performance benefits and measurable ROI) being key points.

Mitigation Strategies

Challenges identified for each of the main facets regarding the effective *design, use,* and *adoption* of AugCog mitigation strategies included:

- User, system, and task context must all be accounted for when determining when and how to apply a mitigation strategy, and systems must be flexible and adaptable enough (and predictive) to meet sometimes rapid and dynamically changing needs of all these contexts, thereby having strategies that are both specialized and generalizable when appropriate.
- Mitigation strategies need to be designed based on solid theoretical and applied foundations (e.g., cognitive, psychology, neuroscience, interface design), where strategies have been proven to be affecting what they should be (e.g., cognitive state, bottlenecks and subsequent user performance) and have been proven to be controllable in terms of potential negative consequences when they are then turned off.
- Traditional user interface and usability concerns still apply, such as consistency, ease of use, smooth state transitions (e.g., mitigation on/off), reliability, user trust, and so forth.
- A methodological approach to both design and implementation of AugCog mitigation strategies is needed for both specialized (e.g., task-specific) and general use conditions.
- Sponsors and investors need to see consistent proof (e.g., experimental data, case studies, field uses, etc.) of how all the above challenges are being understood, met, and overcome and that benefits of applying mitigation strategies can be maximized while costs are minimized in order to justify funding more R&D that is necessary to attain proven, robust and matured mitigation strategies.
- It should be demonstrated how the mitigation strategies are "AugCog" relevant and not just better HCI.

Robust Controllers

The most important challenges to the effective design, use, and adoption of robust controllers were also noted and included the following:

- Given the complexity of HIP, mathematical approaches are needed, but are not the sole answer to developing robust controllers, as there is a need for real-time user and system models based on complex human biology and physiology, as well as a need for task models, which take all contexts into account.
- More basic and applied research is needed to develop sufficiently comprehensive and accurate models of the components (e.g., user, task, system contexts) that are to be controlled, as well as determining how to control them with appropriate approaches, be they mathematical or otherwise.
- Any robust controller should be stable and seamless, where a user is unaware of when a controller is being used yet trusts the system when subsequent effects are noticed (e.g., mitigation turned on/off) and therefore benefits and not suffers from the effects.
- Both users and funding sponsors need proof of the effectiveness of any controller's ability to integrate the human within the system-of-systems architecture, where input and output of information flow and mitigation control are sufficiently adaptable to improve user performance and maintain overall system stability.
- Return on Investment (ROI) must be justified in terms of development cost (time and money) and benefits (significant user/system performance improvements).

Roles of National and Supra-National Institutions

To aid in ensuring that the effective design and implementation of AugCog technologies continues, a few main challenges for national and supra-national institutions (e.g., DARPA, ONR, NIH, and NSF) were proposed. The overarching theme is the need for more funding to be funneled into the development of AugCog sensors and that institutions need to foster the ability of HCI practitioners to begin identifying uses for neurotechnologies in various application domains. Pursuing such goals will eventually increase the competition and foster more development and maturity of AugCog neurotechnologies.

CONCLUSION

The field of AugCog has emerged in large part as a result of substantial investment from DoD-funded programs and projects that have been focused on developing neurophysiological-based tools and techniques to enable revolutionary changes in HCI system design. Whether the application context is an operational close-loop system, system design and evaluation, education and training, or simply entertainment, such tools and techniques offer the ability to create human-machine synergy and optimization never before realized. Similar to Licklider's original visions of human-machine symbiosis in 1960, AugCog researchers and practitioners aim to build tightly coupled brain-machine interfaces that surpass the information-handling capacity of traditional HCI systems and empower one operator with the ability to perform a job normally required of two or more operators. Such an improvement in the human-computer integral is a worthy goal. Being able to noninvasively measure and assess users' cognitive state in real time, and then use automated computational systems to modify and enhance HIP capabilities of these users in any application context, is a goal that could substantially improve human performance and the way humans interact with computer-based systems in the 21st century. It is up to HCI researchers and practitioners to begin implementing various AugCog tools and techniques by selecting the most appropriate neurophysiological tools for their task applications and users. The technologies reviewed in this chapter will hopefully provide a nice "initial" toolset from which to begin such a selection process.

References

Bonner, M. C., Taylor, R. M., Fletcher, K., & Miller, C. (2000). Adaptive automation and decision aiding in the military fast jet domain. *Proceedings of the Conference on Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium.* Savannah, GA, pp. 154–159. Also available on line March 27, 2007 from: http://www.dtic.mil/dticasd/sbir/sbir041/srch/af071.pdf

Ciesielski K. T., & French C. N. (1989). Event-related potentials before and after training: Chronometry and lateralization of visual N1 and N2. *Biological Psychology, 28*(3), 227–238.

Deeny, S. P., Hillman, C. H., Janelle, C. M., & Hatfield, B. D. (2003). Cortico-cortical communication and superior performance in skilled marksmen: An EEG coherence analysis. *Journal of Sport & Exercise Psychology, 25*(2), 188–204.

Dickson, B. T. (2005a). The cognitive cockpit—A testbed for augmented cognition. *CD Proceedings of the First International Conference on Augmented Cognition, 11*, (pp. 479–488). Las Vegas, Nevada.

Dickson, B. T. (2005b). Closed-loop systems—Stability and predictability. *CD Proceedings of the First International Conference on Augmented Cognition, 11*, (pp. 617–620). Las Vegas, Nevada.

Dorneich, M. C., Whitlow, S. D., Mathan, S., Ververs, P. M., Erdogmus, D., Adami, A., Pavel, M., & Lan, T. (2007). Supporting real-time cognitive state classification on a mobile individual. *Journal of Cognitive Engineering and Decision Making* (JCEDM) (special issue), *1* [in press].

DuRousseau, D. R. (2004a). Spatial-frequency patterns of cognition. *The AUGCOG Quarterly, 1*(3), 10.

DuRousseau, D. R. (2004b) *Multimodal cognitive assessment system* (Final technical report, DARPA, DAAH01-03-C-R232). Publisher's location: Arlington, VA: DARPA.

Floyer-Lea, A., & Matthews P. M. (2004). Changing brain networks for visuomotor control with increased movement automaticity. *Journal of Neurophysiology, 92*(4), 2405–2412.

Freeman, F. G., Mikulka, P. J., Prinzel, L. J., & Scerbo, M. W. (1999). Evaluation of an adaptive automation system using three EEG indices with a visual tracking system. *Biological Psychology, 50*, 61–76.

Gevins, A., & Smith, M. E. (2000). Neurophysiological measures of working memory and individual differences in cognitive ability and cognitive style. *Cerebral Cortex, 10*(9), 829–839.

Gratton, G., & Fabiani, M. (2001). The event-related optical signal: A new tool for studying brain function. *International Journal of Psychophysiology, 10*(2), 109– 121.

Gundel, A., & Wilson, G. F. (1992). Topographical changes in the ongoing EEG related to the difficulty of mental tasks. *Brain Topography, 5*(1), 17–25.

Gunter T. C., & Friederici A. D. (1999). Concerning the automaticity of syntactic processing. *Psychophysiology, 36*(1), 126–137.

Hess, E. H., & Polt, J. M. (1964). Pupil size in relation to mental activity during simple problem solving. *Science, 143*, 1190–1192.

Hoover, A., & Muth, E. (2004). A real-time index of vagal activity. *International Journal of Human-Computer Interaction, 17*(2), 197–210.

Humphrey, D. G., & Kramer, A. F. (1994). Toward a psychophysiological assessment of dynamic changes in mental workload. *Human Factors, 36*(1), 3–26.

Just, M. A., & Carpenter, P. A. (1993). The intensity dimension of thought: Pupillometric indices of sentence processing. *Canadian Journal of Experimental Psychology, 47*(2), 310–339.

Kass, S. J., Doyle, M., Raj, A. K., Andrasik, F., & Higgins, J. (2003). Intelligent adaptive automation for safer work environments. In J. C. Wallace & G. Chen (Chairs), *Occupational health and safety: encompassing personality, emotion, teams, and automation*. Symposium conducted at the Society for Industrial and Organizational Psychology 18th Annual Conference, Orlando, FL.

Kerick, S. E., Douglass, L. W., & Hatfield, B. D. (2004). Cerebral cortical adaptations associated with visuomotor practice. *Med Sci Sports Exercise, 36*(1), 118–129.

Kobus, D. A., Brown, C. M., Morrison, J. G., Kollmorgen, G., & Cornwall, R. (2005, October 31). DARPA Improving Warfighter Information Intake under Stress—Augmented Cognition Phase II: The Concept Validation Experiment (CVE). DARPA/IPTO technical report submitted to CDR Dylan Schmorrow.

Kollmorgen, L. S. (2007). Introduction: A case for operational approach in advanced research projects—the augmented cognition story. *Aviation Space and Environmental Medicine* (Special Supplement), *78*(5) [in press]..

Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics, 1,* 4–11.

Mikulka, P. J., Scerbo, M. W., & Freeman, F. G. (2002). Effects of a biocybernetic system on vigilance performance. *Human Factors, 44*(4), 654–664.

Parasuraman, R. (2003). Neuroergonomics: Research and practice. *Theoretical Issues in Ergonomics Science, 4*, 5–20.

Peres, M., Van De Moortele, P. F., Pierard, C., Lehericy, S., Satabin, P., Le Bihan, D., et al. (2000). Functional magnetic resonance imaging of

mental strategy in a simulated aviation performance task. *Journal of Aviation Space Environmental Medicine, 71*(12), 1218–1231.

Pleydell-Pearce, K., Dickson, B., & Whitecross, S. (2000). Cognition monitor: A system for real time pilot state assessment. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp. 65–69). London: Taylor & Francis.

Pope, A. T., Bogart, E. H., & Bartolome, D. S (1995). Biocybernetic system validates index of operator engagement in automated task. *Biological Psychology, 40*, 187–195.

Prinzel, L. J., Hadley, G., Freeman, F. G., & Mikulka, P. J. (1999). Behavioral, subjective, and psychophysiological correlates of various schedules of short-cycle automation. In M. Scerbo & K. Krahl (Eds.), *Automation technology and human performance: Current research and trends*. Mahwah, NJ: Lawrence Erlbaum Associates.

Proffit, D. (2003, March 5). *Human information processing bottlenecks.* Paper presented at the Augmented Cognition Technology Integration Event, San Diego, CA.

Raj, A. K., Perry, J. F., Abraham, L. J., & Rupert A. H. (2003). Tactile interfaces for decision making support under high workload conditions. *Proceedings of Aerospace Medical Association 74th Annual Scientific Meeting*, San Antonio, TX.

Reeves, L. M., Schmorrow, D. D., & Stanney, K. M. (2007). Augmented cognition and cognitive state assessment technology—near-term, mid-term, and long-term research objectives. In *Proceedings of the Augmented Cognition International Conference* [in press]. Beijing, China.

Reeves, L. M., Young, P., Schmorrow, D. D., & Stanney, K. M. (2007). Augmented cognition and robust controller technology—near-term, mid-term, and long-term research objectives. In *Proceedings of the Augmented Cognition International Conference* [in preparation]. Baltimore, MD.

Sanders, M. S., & McCormick, E. J. (1993). *Human factors in engineering and design* (7th ed.). New York: McGraw-Hill.

Sajda, P., Gerson, A., & Parra, L. (2003). Spatial signatures of visual object recognition events learned from single-trial analysis of EEG. *Proceedings of IEEE Engineering in Medicine and Biology Annual Meeting* (pp. 2087–2090). Cancun, Mexico.

Schmorrow, D., & Kruse, A. (2002). Improving human performance through advanced cognitive system technology. In CD *Proceedings of the Interservice/Industry Training, Simulation & Education Annual Conference (I/ITSEC'02)*. Orlando, FL.

Schmorrow, D. D., & Kruse, A. A. (2004). Augmented cognition. In W. S. Bainbridge (Ed.), *Berkshire encyclopedia of human-computer interaction* (pp. 54–59). Great Barrington, MA: Berkshire Publishing Group.

Schmorrow, D., & Kruse, A. (2005). Session overview: Foundations of augmented cognition. In D. D. Schmorrow (Ed.), *Foundations of Augmented Cognition* (pp. 441–445). Mahwah, NJ: Lawrence Erlbaum Associates.

Schmorrow, D., & McBride, D. (2004). Introduction, Augmented Cognition [Special issue]. *International Journal of Human-Computer Interaction, 17*(2), 127–130.

Schmorrow, D., Stanney, K. M., Wilson, G., & Young, P. (2005). Augmented cognition in human-system interaction. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (3rd ed.). New York, NY: John Wiley.

Schmorrow, D. D. & Reeves, L. M. (2007). Introduction: 21st century human-system computing: augmented cognition for improved human performance. *Aviation Space and Environmental Medicine* (Special Supplement), *78*(5) [in press].

Smith, E. E., & Jonides, J. (1998). Neuroimaging analyses of human working memory. *Proceedings of the National Academy of Sciences 95*(20), 12061–12068. USA.

Smith M. E., McEvoy L. K., & Gevins A. (1999). Neurophysiological indices of strategy development and skill acquisition. *Cognitive Brain Research, 7*(3), 389–404.

Taylor, R. M., Howells, H., & Watson, D. (2000). The cognitive cockpit: Operational requirement and technical challenge. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp. 55–59). London: Taylor & Francis.

Wickens, C. D., Heffley, E., Kramer, A. & Donchin, E. (1980). The event-related brain potential as an index of attention allocation in complex displays. *Proceedings of the 24th Annual Meeting of the Human Factors Society* (pp. 297–301). Santa Monica, CA.

Wilson, G. F., & Schlegel, R. E. (Eds.). (2004). *Operator functional state assessment* (NATO RTO Publication RTO-TR-HFM-104). Neuilly sur Seine, France: NATO Research and Technology Organization.

·63·

HUMAN VALUES, ETHICS, AND DESIGN

Batya Friedman and Peter H. Kahn, Jr.
University of Washington

Introduction **1242**
How Values Become Implicated in Technological Design **1242**
The Embodied Position 1242
The Exogenous Position 1243
The Interactional Position 1243
Distinguishing Usability from Human Values with Ethical Import **1244**
Moral, Philosophical, and Psychological Foundations **1245**
Moral Theory 1245
Moral Epistemology 1246
Moral Variability and Universality 1246
Designing for Diversity 1247
Approaches to Human Values and Ethics in Design **1247**
Computer Ethics 1247
Social Informatics 1248
Computer-Supported Cooperative Work 1249
Participatory Design 1250
Value Sensitive Design 1250
Human Values with Ethical Import **1251**
Human Welfare 1252
Ownership and Property 1252
Privacy 1252
Freedom from Bias 1253
Universal Usability 1253
Trust 1254
Autonomy 1254
Informed Consent 1255
Accountability 1255
Identity 1256
Calmness 1256
Environmental Sustainability 1257
Professional Ethics **1257**
Conclusions **1259**
Acknowledgments **1260**
References **1261**

INTRODUCTION

Human values and ethical considerations no longer stand apart from the human-computer interaction (HCI) community—perhaps in some separate field called "computer ethics"—but are fundamentally part of our practice.

This shift reflects, at least in part, the increasing impact and visibility that computer technologies have had on human lives. Computer viruses have destroyed data on millions of machines. Large linked medical databases can, and often do, infringe on individuals' privacy. Accountability becomes a problem when large computer systems malfunction and result in human deaths. The outcome of the 2000 presidential elections in the United States may have hinged on the poor design of the computerized Florida election ballot (the "butterfly ballot"). On and on, the media portray such problems. Presidential reports (PITAC, 1999) ask that such problems be addressed. Corresponding research agendas on human values have been integrated into recent government-funded programs, such as the National Science Foundation's funding initiative on information technology research. Thus, the writing decades ago by cyberneticists like Norbert Wiener (1953/1985) and computer scientists like Joseph Weizenbaum (1972) now seem prescient. They argued that humans fundamentally control technology and that we must make wise and humane choices about its design and use.

But, if human values, especially those with ethical import—such as rights to privacy and property, physical welfare, informed consent, trust, and accountability, to name but a few—are important, they are no less controversial. What values count? Who decides? Are values relative, or are some values universal but expressed differently across cultures and contexts? Does technology even have values? If not, how do values become implicated in design? It is also clear that values can conflict. For example, shared online calendars can enhance access to information within an organization but at the expense of individual privacy (Palen & Grudin, 2002). Thus, on what basis do some values override others in the design of, say, hardware, algorithms, databases, and interfaces? Finally, how can HCI designers working within a corporate structure and with a mandate to generate revenue bring values and ethics into their designs?

In this chapter, we review how the field has addressed such questions. The second section reviews positions on how values become implicated in technological designs. The third section distinguishes usability from human values with ethical import. The fourth section reviews foundational issues that make ethics so controversial. We also show how scholars have bounded moral controversies in ways that make it possible for practitioners to shape their work from an ethical stance. The fifth section reviews the major HCI approaches to the study of human values, ethics, and design: computer ethics, social informatics, computer-supported co-operative work, participatory design, and value sensitive design. We discuss the strengths (and limitations) of each approach by drawing on exemplar projects. The sixth section calls attention to key values relevant for design: human welfare, ownership, property, privacy, freedom from bias, universal usability, trust, autonomy, informed consent, accountability, identity, calmness, and environmental sustainability. Each of these values could merit its own chapter; and, indeed, in this handbook, some do. Yet by reviewing them here, however briefly, we highlight their ethical status and thereby suggest that they have a distinctive claim on resources in the design process. The seventh section reviews the special ethical responsibilities that HCI professionals incur by virtue of their professional standings. Finally, in the eighth section, we highlight a few suggestions for moving the field forward.

HOW VALUES BECOME IMPLICATED IN TECHNOLOGICAL DESIGN

Technological innovations implicate human values. Consider, for example, that in the early 1900s, missionaries introduced a technological innovation—steel ax heads—to the Yir Yoront of Australia, a native people. The missionaries did so without regard for traditional restrictions on ownership and indiscriminately distributed the ax heads to men, women, old people, and young adults alike. In so doing, they altered relationships of dependence among family members and reshaped conceptions of property within the culture (Sharp, 1952/1980). In another example, about four decades ago, snowmobiles were introduced into the Inuit communities of the Arctic and largely replaced travel by dog sleds. This technological innovation thereby altered not only patterns of transportation, but also symbols of social status, and moved the Inuit toward a dependence on a money economy (Houston, 1995; Pelto, 1973). In a computer example, electronic mail rarely displays the sender's status. Is the sender perhaps a curious layperson, a system analyst, a full professor, a journalist, an assistant professor, an entry-level programmer, a senior scientist, or a high-school student? Nobody knows until the e-mail is read (and maybe not even then). This design feature (and associated conventions) has thereby played a significant role in allowing electronic communication to cross traditional, hierarchical boundaries and to contribute to the restructuring of organizations (Sproull & Kiesler, 1991).

However, how exactly do values become implicated in technological designs? Three types of positions have been offered in the literature: embodied, exogenous, and interactional.

The Embodied Position

The embodied position holds that designers inscribe their own intentions and values into the technology; once developed and deployed, the resulting technology determines specific kinds of human behavior. Indeed, in the history of science literature (Smith & Marx, 1994), this position is sometimes referred to as "technological determinism." To illustrate the idea, consider Latour's (1992) description of the "Berliner lock and key": a device designed to deal with people who forget to lock the door behind them. The device initially works as a normal lock. You put your key into the lock, turn the key, and the door unlocks. Then, to remove the key, it is necessary to push it through the key hole to the other side. Then, by moving to the other side of the door, you rotate the key one more turn, thereby relock-

ing the door. Thus, when used, this technology determines that people will lock their doors.

There is a hard and soft version of the embodied position. In the hard version, it is argued that the very meaning and intentions that designers and builders bring to their task literally become a part of the technology (cf., Appadurai, 1988; Cole, 1991). Many people have trouble with this version because it imputes mental states to things that do not seem to have the capacity to have them. As Smith and Marx (1994) wrote:

> Critics of "hard" determinism question the plausibility of imputing agency to "technology". . . . How can we reasonably think of this abstract, disembodied, quasi-metaphysical entity [technology], or of one of its artifactual stand-ins (e.g., the computer), as the initiator of actions capable of controlling human destiny? (p. xii)

Yet, it is possible that as the field of artificial intelligence moves forward—for example, as computational learning systems increasingly mimic human agency—the strong version of the embodied position will gain some currency.

In the more common soft version, it is recognized that objects themselves do not literally embody an intention or value. It is also recognized that designers themselves are shaped by organizational, political, and economic forces and that, because of such forces, a particular technology may never take hold in a society. But, it is argued, if the technology takes hold, then it becomes very difficult for an individual or society to override the values driven by the technology. Think about how the design and deployment of passenger-side front air bags have in a sense held children hostage in the back seat of millions of cars. Or, consider that a virus program readily destroys data or grabs CPU cycles (or whatever it is programmed to do). Granted, you could perhaps use the program to teach about self-replicating code, but such activity would in effect be secondary to its primary function. In other words, to execute the computer virus code likely leads to the destruction of data. Similarly, if your goal in online interactions is to keep data secure, you can design technical security mechanisms, such as mandatory encryption, certificates, and anonymizers. Such designs ensure behavior. Thus, according to Akrich (1992), a large part of a designer's work is that of "inscribing" or embodying his or her vision of "the world in the technical content of the new object" (p. 208).

The Exogenous Position

The exogenous position holds that societal forces—that involve, for example, economics, politics, race, class, gender, and religion—significantly shape how a deployed technology will be used. Consider, for example, the Internet. In many ways, its basic file-sharing functionality remains unchanged from its initial inception. Files are still broken into packets, and packets are routed and then reassembled. Why is it, then, that this underlying technology was primarily used for nearly a quarter of a century by the scientific and educational communities before becoming appropriated by the private sectors for commerce? Once appropriated, why did the development of e-commerce occur so rapidly? According to the exogenous theorists, answers to such questions cannot be found in the supposedly embodied values of the technology itself. Rather:

> To understand the origin of a particular kind of technological power, we must first learn about the actors. Who were they? What were their circumstances? . . . Why was this innovation made by these people and not others? Why was it possible at this time and this place rather than another time and place? . . . Instead of treating "technology" per se as the locus of historical agency . . . [these advocates] locate it in a far more various and complex, social, economic, political and cultural matrix. (Smith & Marx, 1994, p. xiii)

In turn, the exogenous position has been used and developed in various ways. For example, Bulliet (1994) examined three technologies that emerged in the Islamic world in the fifth century: block printing, the harnessing of draft animals, and wheeled transport. Each technology failed to have an immediate transforming social and economic impact, not because of its lack of economic advantage, but because of social filters, which involved race, class, and lifestyle. Perdue (1994) explored the comparative history of agrarian societies (medieval Western European, 18th-century Russian, and 14th-century Chinese). He argued that studies of technology need contextual accounts that integrate environmental, technological, social, and cultural elements. Scranton (1994) drew on postmodern theory to argue that local contingencies, diversities, disjunctions, multiple oppositions, and contrasting norms should be central to exogenous explanations. Hughes (1994) suggested that younger developing technologies tend to be more open to socio-cultural influences, whereas older, more mature technologies tend to be more deterministic. Through many such studies, a recurring view is that technological systems are not value neutral but invariably favor the interests of people with economic and political power (Noble, 1991; Smith, 1994; Winner, 1986).

The Interactional Position

The interactional position holds that, whereas the features or properties that people design into technologies more readily support certain values and hinder others, the technology's actual use depends on the goals of the people interacting with it. A screwdriver, after all, is well suited for turning screws, and yet amenable as a poker, pry bar, nail set, cutting device, and tool to dig up weeds. Moreover, through human interaction, technology itself changes over time. On occasion, such changes (as emphasized in the exogenous position) can mean the societal rejection of a technology, or that its acceptance is delayed. But more often, it entails an iterative process whereby technologies get invented and then redesigned based on user interactions, which then are reintroduced to users, further interactions occur, and further redesigns implemented. Typical software updates (e.g., of word processors, browsers, and operating systems) epitomize this iterative process.

Two sides of the interactional position have been emphasized in the literature. One side emphasizes the properties designed into the technology (Friedman, 1997a). For example, for

the moment, let us agree that disabled people in the work place should be able to access technology, just as they should be able to access a public building (Perry, Macken, Scott, & McKinley, 1997). As system designers, we can make the choice to try to construct a technological infrastructure that disabled people can access. If we do not make this choice, then we single handedly undermine the human value of universal access.

The other side of the interactional position emphasizes how users use the technology in the context of social-organizational structures. Orlikowski (2000), for example, studied the use of Lotus Development Corporation's *Notes* software by two groups within a large multinational consulting firm: technologists and consultants. The technologists used *Notes* extensively. They used e-mail, maintained electronic discussions with *Notes* databases, and created their own database designs. Moreover, "supported by the cooperative norms of technical support, the technologists used many of the properties of *Notes* to promote their collective technical work, and to cooperate with each other" (Orlikowski, 2000, p. 415). In contrast, the consultants used *Notes* minimally, sometimes even begrudgingly. Orlikowski found that what she calls such "technology-in-practice" was enacted for at least three different reasons. First, some consultants had doubts about the value of *Notes* for their performances. Second, in contrast to the technologists, the consultants were under a time-based billing structure. "Because many consultants did not see using *Notes* as an activity that could be billed to clients, they were unwilling to spend time learning or using it" (Orlikowski, 2000, p. 416). Third, consultants feared that the collaborative properties of *Notes* would threaten their statuses within the company. Thus, Orlikowski proposed "a view of technology structures, not as embodied in given technological artifacts, but as enacted by the recurrent social practices of a community of users" (p. 421).

Regardless of emphasis, from the interactional position it should be clear that design and social context matter, dialectically. Moreover, users are not always powerless when faced with unwelcome value-oriented features of a technology.

DISTINGUISHING USABILITY FROM HUMAN VALUES WITH ETHICAL IMPORT

The language and conceptualizations within HCI currently provide solid means by which to pursue issues of usability (Adler & Winograd, 1992; Nielsen, 1993; Norman, 1988). Usability refers to characteristics of a system that make it work in a functional sense, including that it is easy to use, easy to learn, consistent, and recovers easily from errors.

Some HCI designers, however, have a tendency to conflate usability with human values with ethical import. (In the next section, we discuss what exactly is meant by the term *moral.* Here, it can be understood to involve issues of fairness, justice, human welfare, and virtue.) This conflation arises insofar as usability is itself a human value although not always a moral one. However, when it is, both can be addressed by the same design. For example, systems that can be modified by users to meet the needs of specific individuals or organizations can both (a) enhance usability and (b) help users to realize their goals and intentions: the moral value of autonomy. Other times, however, usability can conflict with human values with ethical import. Nielsen (1993), for example, asked us to imagine a computer system that checks for fraudulent applications of people who are applying for unemployment benefits. Specifically, the system asks applicants numerous personal questions, and then checks for inconsistencies in their responses. Nielsen's point is that even if the system receives high usability scores, some people may not find the system socially acceptable, based on the moral value of privacy.

In terms of a general framework, four pair wise relationships exist between usability and human values with ethical import.

1. *A design is good for usability and independently good for human values with ethical import*. This relationship is exemplified previously, where adaptable systems can also promote user autonomy. Another example involves browser designs that offer users more efficient cookie management than currently offered in Netscape or Internet Explorer and that promote values of informed consent and privacy (Friedman, Howe, & Felten, 2002).
2. *A design is good for usability, but at the expense of human values with ethical import*. This relationship is exemplified in the previous example offered by Nielsen (1993), where a highly usable system undermines the value of privacy. Tang (1997) provided another example by means of a case study of a team designing a workstation. At one point, the team was trying to decide how to power a microphone, and they finally decided to power it directly from the workstation. Consequently, they eliminated a separate hardware on/off switch on the microphone. From a usability perspective, Tang pointed to benefits of this design. Users, for example, no longer had to remember to turn the microphone off when it was not in use (to conserve battery power) or be inconvenienced by replacing a dead battery. However, despite these usability advantages, Tang found that some users thought the design—by not allowing them direct control over the microphone—undermined their privacy and security.
3. *A design is good for human values with ethical import, but at the expense of usability*. This relationship is exemplified by choosing a browser preference with the setting "accept or decline each cookie individually." The values of privacy and informed consent are well supported, but, for most people, the nuisance factor is too high. Another example (the flip side of an earlier one) arises when a system is purposefully made adaptable to promote autonomy, but in the process, the system becomes unwieldy and difficult to use.
4. *A design good for usability is necessary to support human values with ethical import*. This relationship is exemplified by security systems that are not so cumbersome to use that either critical features of the security system are disabled or individuals invent workarounds that compromise the security. For example, if a security system requires three passwords and 10 steps to login, a user's workaround might involve taping a yellow sticky to the side of the computer screen with the passwords (for an in depth treatment of usable security, see Cranor & Garfinkel, 2005). Other instances of this relationship arise when morality requires that all individuals in a specified group be able to use the system suc-

cessfully. For example, to have a fair national election using a computerized voting system, all citizens of voting age must be able to use the system. Bederson, Lee, Sherman, Herrnson, and Niemi (2003) surveyed issues relating to usability of electronic voting system and reported voter concerns.

HCI professionals are often responsible for usability. Accordingly, it is important that we be aware of the possible relationships between usability and human values with ethical import (e.g., Cranor & Garfinkel, 2005, on security and usability). At times, the two support one another. Other times we need to give ground on usability to promote human values with ethical import or, conversely, give ground on human values with ethical import to promote usability. Such optimizations require judicious decisions, carefully weighing and coordinating the advantages of each.

MORAL, PHILOSOPHICAL, AND PSYCHOLOGICAL FOUNDATIONS

HCI professionals—like people in many fields—may sometimes wonder if morality is too controversial to be integrated in a principled way into their work. After all, who is to say for another person what is right, wrong, good, or bad? How do we make sense of the seemingly different moralities among people? Is not morality relative to person or at least culture?

Such questions form the backdrop to a wide range of issues that moral philosophers and psychologists have pursued. In this section, we review relevant literature to help bound moral controversies such that HCI researchers and practitioners can, with legitimate grounding, move forward proactively in shaping their work from a moral stance.

Moral Theory

One common starting point in moral discussions is to raise the questions, "What do you mean by morality?" and "How do you define it?" In response, moral philosophers have offered what can be viewed as three overarching approaches toward developing moral theory: consequentialist, deontological, and virtue-based. Briefly stated, consequentialist theories maintain that a moral agent must always act to produce the best available outcomes overall (for an analysis, see Scheffler, 1982). Utilitarianism is the most common form of consequentialism, wherein a moral agent should act to bring about the greatest amount of utility (e.g., happiness) for the greatest number of people. In contrast, deontological theories maintain that a moral agent is forbidden to do some actions, or in turn, must do, regardless of consequences (Dworkin, 1978; Gewirth, 1978; Kant, 1785/1964; Rawls, 1971). For example, a deontologist might maintain that it is immoral to torture an innocent child even if such an act would bring about great good (e.g., to prevent a bombing). Along similar lines, it can be argued that vendors who collect and sell information about individuals' web-browsing activity without individuals' knowledge and consent violate individuals' right to privacy. Indeed, most any argument for a basic right makes a deontological move.

Both consequentialist and deontological theories are centrally concerned with answering the fundamental question, "What ought I to do?" In turn, answers are often viewed to be morally obligatory, meaning that the action is required of every moral agent "regardless of whether he wants to accept them or their results, and regardless also of the requirements of any other institutions such as law or etiquette" (Gewirth, 1978, p. 1). Moreover, such obligatory prescriptions are often framed in the negative (as in "Thou shall not steal") such that it is possible to fulfill the prescription (e.g., just do not steal). In turn, virtue-based theories are centrally concerned with answering the fundamental question, "What sort of person ought I to be?" in which the focus is on long-term character traits and personality (see Louden, 1984). This tradition dates back to Aristotle's delineation in *Nichomachean Ethics* of the ethical virtues (e.g., courage, temperance, friendship, wisdom, and justice), and developed, for instance, by Foot (1978), MacIntyre (1984), and Campbell and Christopher (1996). In virtue ethics, the prescription is often viewed to be morally discretionary, meaning that whereas an action is not morally required, it is conceived of as morally worthy and admirable based on considerations of welfare and virtue (Kahn, 1992). Moreover, such discretionary prescriptions are often framed in the positive (as in "be charitable"), such that it is not possible to satisfy the declaration completely. After all, one cannot continuously practice charity without soon becoming destitute oneself (Fishkin, 1982).

With these core distinctions in mind—between a theory of the right (both consequentialism and deontology) that is largely viewed as obligatory and a theory of the good (virtue theory) that is largely viewed to be discretionary—it is possible to characterize a wide range of HCI designs. For example, consider Mattel Corporation's "Barbie doll" website that was up and running during the latest census collection in the United States. On the website, Barbie posed as a census taker and asked children to provide information about their families. Presumably, Mattel sought to use the resulting information to market their products more effectively. Is this action moral? Most people would presumably say no. But why? From a deontological position, such a deceptive web design violates a moral obligation (e.g., the obligation not to intentionally deceive others, let alone to deceive children for a corporation's material gain), and should not have been designed and deployed.

Other human values articulated in HCI fit within a virtue orientation. For example, SeniorNet is an organization that brings seniors together via computer networks. Mynatt, Adler, Ito, Linde, and O'Day (1999) found that "SeniorNetters repeatedly commented on the warmth and friendliness of the community as something that differentiated SeniorNet from other net communities, and as a reason for their participation and comfort with the community" (p. 232). Are values such as warmth and friendliness moral values? For a consequentialist and even more so for a deontologist, the answer is probably no—a person, after all, is presumably not under a moral obligation to engage in warm-hearted actions. However, for a virtue ethicist the answer is fundamentally yes, as such values are conceived to be central to moral personhood. Consider websites (such as Alfie .com) that offer children a menu of computer games, like pinball or miniature golf, that children can play. Most of these games do not promote such virtues as friendship, caring, or

compassion. Must they? Presumably, even virtue theorists would say that not every game has to. Rather, promoting such virtues is usually conceived as discretionary—perhaps morally praiseworthy if done, and perhaps contributing to an account of human flourishing—but not morally obligatory.

Moral Epistemology

This discussion raises the question, "Who is to say what's moral?" or "How do we know?" Such questions move us into the field of moral epistemology: the study of the limits and validity of moral knowledge. Often at stake is whether it is even possible for a moral statement to be objectively true or false and for a moral value to be objectively right, wrong, good, or bad. A wide variety of positions has been taken. For instance, some believe that moral knowledge corresponds to or approaches a correspondence with a moral reality that exists independent of human means of knowing (Boyd, 1988; Sturgeon, 1988). Others believe moral knowledge can be objectively grounded by constructing rational principles that strive for coherence and consistency while building on the common ground and specific circumstances of a society (Dworkin, 1978; Habermas, 1979). Others believe the only thing that can be said of moral knowledge is that it can be true subjectively for an individual or culture depending on that individual's or group of individuals' desires, preferences, and goals (Mackie, 1977; Rorty, 1982). Finally, others believe that any moral knowledge is unattainable, even in a weak sense (the full skeptic's position; for a characterization, see Nagel, 1986).

The skeptic's position—that moral knowledge is unattainable—reflects the morally relativistic position described previously: that no one can say what is really right or wrong, so ethics and values become, at best, a personal choice in terms of HCI design. However, consider that no one can prove that at this moment you are not really just a brain in a vat being stimulated by electrical impulses to think that you are reading this chapter. Still, you have compelling reasons to think that proposition false and similarly with morality. Although the skeptic's position cannot be proved conclusively false, moral philosophers have provided compelling reasons not to believe it (Nagel, 1986; Williams, 1985).

Moral Variability and Universality

Anthropologists often document moral differences between cultures. For example, from some anthropological accounts, we learn that devout Hindus believe that it is immoral for a widow to eat fish or for a menstruating woman to sleep in the same bed with her husband (Shweder, Mahapatra, & Miller, 1987). Other accounts document that members of the Yanomamo tribe of Brazil at times practice infanticide and that the women are "occasionally beaten, shot with barbed arrows, chopped with machetes or axes, and burned with firebrands" (Hatch, 1983, p. 91). Per our discussion, there are three ideas to understand about such examples.

1. *Variability in human practice does not prove or disprove the moral skeptic's position*. Imagine going to a culture where the people there did not believe in logical transitivity. (If $A = B$ and $B = C$, then $A = C$.) That finding would be interesting psychologically and culturally, but presumably has no bearing on whether logical transitivity is true or false. It is or it is not. If it is true, then people who think otherwise are simply wrong. So, too, with the moral life. People can believe that a certain act (such as shooting women with barbed arrows) is moral; but documenting such a belief does not make it so.
2. *When moral differences occur between peoples, it is not necessarily the case that the practices are believed legitimate by the victims*. For example, in Hatch's (1983) report on the Yanomamo, he also reported that the women did not appear to enjoy such physically abusive treatment and were seen running in apparent fear from such assaults. Psychological data of a similar kind can be found in a study by Wainryb (1995) on the Druze population in Israel. The Druze largely live in segregated villages, are of Islamic religious orientation, and are organized socially around patriarchal relationships. The father, as well as brothers, uncles, and other male relatives, and eventually a woman's husband, exercise considerable authority over women and girls in the family and restrict their activities to a large degree. However, when these women were interviewed, a majority of them (78%) unequivocally stated that the husband or father's demands and restrictions were unfair. Thus, Yanomamo and Druze women—like many women in Western societies—are often unwilling victims within what they themselves perceive to be an uncaring or unjust society. In such situations, it may be less the case that societies differ on moral ground and more that some societies (including Western societies) are involved explicitly in immoral practices (Wainryb, 2006).
3. *Moral variability may be much less pervasive than many people suppose*. Reconsider, for example, the Shweder et al. (1987) report that devout Hindus believe it is immoral for a widow to eat fish or for a menstruating woman to sleep in the same bed as her husband (but two of many dozens of their examples). At first glance, for Western eyes at least, such moral beliefs do seem different. However, when Shweder et al.'s data were reanalyzed by Turiel, Killen, and Helwig (1987), they showed bases for not only difference but also moral commonality. For example, in their reanalysis, Turiel et al. found that devout Hindus believed that harmful consequences would follow from a widow who ate fish (the act would offend her husband's spirit and cause the widow to suffer greatly) and from a menstruating woman who sleeps in the same bed with her husband (the menstrual blood is believed poisonous and can hurt the husband). Although such beliefs, themselves, differ from those in Western culture, the underlying concern for the welfare of others is congruent with them. More generally, Turiel et al. claimed that when researchers differentiate informational and metaphysical assumptions about the world from moral judgments based on those assumptions, then the moral life often takes on a greater universal cast (cf., Helwig, 2006; Turiel, 1998, 2002).

When analyzing moral variability, conceptualizations of morality that entail abstract characterizations of justice and welfare tend to highlight moral universals, whereas definitions that

entail specific behaviors or rigid moral rules tend to highlight moral cross-cultural variation. Typically, theorists who strive to uncover moral universals believe they are wrestling with the essence of morality, with its deepest and most meaningful attributes. In contrast, theorists who strive for characterizing moral variation argue that, by the time a person has a common moral feature that cuts across cultures, he or she has so disembodied the idea into an abstract form that it loses virtually all meaning and utility. For instance, in the example of devout Hindus who believe that by eating fish a widow hurts her dead husband's spirit, is the interesting moral phenomenon that Hindus, like ourselves, are concerned with not causing others harm? Or, is the interesting moral phenomenon that Hindus believe in spirits that can be harmed by earthly activity?

Both questions have merit, and a middle ground provides a more sensible and powerful approach for the HCI community: one that allows for an analysis of universal moral values, as well as allowing for these values to play out differently in a particular culture at a particular point in time (Friedman, 1997b; Kahn, 1991, 1999; Kahn & Lourenco, 1999).

Designing for Diversity

In this chapter, we will continue to draw on the preceding analysis to help us review the HCI literature and to offer morally principled design methods that respect culture and context. As a case in point, imagine you are designing a computerized voting booth. At what height would you place the electronic ballot? A reasonable answer might go something like "It depends. How tall are the people who vote? Moreover, later, will the voting booth be used by other people of different height? After all, a voting booth designed only for players of the National Basketball Association will disenfranchise most voters in Japan." In other words, the universal value is to enfranchise all voters, but the specific mechanism by which to do so may need to be adaptable to specific contexts and cultures. The general principle then is that designs need to be robust enough to substantiate the value under consideration and yet adaptable enough so that different cultures (or subcultures) can use the designs in their own ways.

Many problems occur when this principle is ignored or when unanticipated users of a system emerge. For example, as described by Friedman and Nissenbaum (1996), since the early 1970s, the computerized National Resident Medical Match Program has placed most medical students in their first jobs. In the system's original design, it was assumed that only one individual in a family would be looking for a residency. At the time, such an assumption was perhaps not out of line, because there were few women residents. However, as women have increasingly made their way into the medical profession, marriages between residents have become more frequent, and bias against placing couples emerged. Another example involved a dog icon that was used to indicate the printing orientation of a printer: landscape or vertical. When the printer was shipped to Islamic countries, the vendor discovered that people in such countries often considered dogs as offensive animals. Consider the case of competitive educational software that was exported to Micronesia, with dismal results because the value of competition promoted by the software clashed too strongly with the culture's emphasis on co-operation. Consider also that data protection laws and policies differ across national boundaries (Agre & Rotenberg, 1998; Rotenberg, 2000). For example, Bennett (1998) described how most of the European countries have applied the same data protection policies to both the public and the private sectors. However, the United States, Canada, Australia, and Japan have preferred "to regulate only the public sector's practices and to leave the private sector governed by a few sectoral laws and voluntary codes of practice" (Bennett, 1998, p. 100). Thus, lenient designs from the private sector of these latter countries need to be adaptable to transfer readily to European countries.

Granted, building value adaptability into systems requires additional time and financial resources. Also, from a user's perspective, additional options add further complexity and challenges. Thus, value adaptability is not always the best way to go. However, it often is, especially when we can anticipate that similar values will play out in different ways for different users. In such cases, not only are a larger number of human lives enhanced from an ethical standpoint but also from an economic standpoint such systems increase market share and generate profits.

APPROACHES TO HUMAN VALUES AND ETHICS IN DESIGN

The computing field has addressed issues of human values and ethics by means of a handful of approaches. To some extent, these approaches overlap with one another. For example, sociotechnical analyses, which are central to social informatics, often form the front end of efforts in participatory design, and are incorporated into the empirical investigations of value sensitive design. Both computer-supported co-operative work and participatory design share particular interests in collaboration in the workplace. However, that said, the approaches differ significantly when considering how each integrates its respective position on moral epistemology, methods, and contexts studied.

In this section, our goal is not to review each approach comprehensively (which is beyond the scope of this chapter). Rather, we seek to show how each approach contributes to our understanding of how to integrate human values and ethics in design.

Computer Ethics

When applied moral philosophers brought their talents and energies to bear on understanding the impact of computer technologies on social life, the field of computer ethics was born (Bynum, 1985; Johnson, 1985; Moor, 1985).

In their resulting examinations, computer ethicists have embraced two complimentary goals. One goal has been to utilize existing moral theory to bring clarity to issues at hand, and—at appropriate times—to proscribe norms of behavior. For example, in the computer science literature, the term *trust* has often been used synonymously with *security* (Schneider, 1999). Yet, drawing on ethical theory, Nissenbaum (1999) showed that these two terms need to be distinguished. For example, if as

HCI professionals, our goal is to create a place where people feel safe in their online interactions, we can move in two design directions. We can move forward with technical solutions that involve security features like locks, keys, passwords, and encryption, or we can understand how trusting relationships are created and fostered, and design them into our online systems. Each direction leads to a very different online environment. The strength of this type of philosophical contribution is that it helps translate moral abstractions into crisp working conceptualizations that HCI professionals can use.

A second goal of computer ethicists builds on the innovations of the technology itself. For the technologies have not only generated new entities (computer programs, the Internet, web pages), but have also enlarged the scale of activities (data mining), increased the power and pervasiveness of its effects (ease of communication), and often became invisible to human purview (using computers for surveillance). Thus, computer ethicists have been interested in understanding how such innovations extend the boundaries of traditional ethical concepts. For example, Moor (1985) examined the ways in which the invisibility of computers affects human lives. "Invisible abuse" is the intentional use of the invisible operations of a computer to engage in unethical conduct, such as the invasion of the property and privacy of others. "Invisible programming values" are those values that are embedded in a computer program, such as bias. "Invisible complex calculations" refer to calculations that are too complex for human inspection and understanding. In all three situations, Moor argued that computer technologies raise special ethical issues different from other forms of technology and that the task of computer ethics is to fill in what he referred to as the resulting "conceptual vacuum" and "policy vacuum" (Moor, 1985, p. 266). In a similar vein, Nissenbaum (2004) argued that some traditional theoretical approaches to privacy yield unsatisfactory conclusions in the case of public surveillance. She posits a new construct, contextual integrity—that demands information gathering and dissemination be appropriate to the context—to capture the nature of challenges posed by information technologies.

One theoretical debate in this literature is whether the technological innovations fundamentally challenge ethical theory (Johnson & Miller, 1997; Spinello & Tavani, 2001). In other words, are innovations so qualitatively different and far reaching from past ones that traditional ethical theories have to be at least significantly revised, if not abandoned? Or, is it the case that the technology simply offers a new domain within which traditional ethical theory works? Toward understanding this debate, a helpful analogy can be made to literature, wherein there exist the categories (genres) of fiction and nonfiction. These categories seem clear enough until we encounter a new form of writing that blurs the boundaries, such as historical novels. Does that new form undermine the traditional categories? Philosophers like Searle (1983) argued no: The categories of fiction and nonfiction fundamentally remain, and what the new literary form shows is that writing can embody aspects of both categories. Similarly, it can be argued that, whereas the technology can blur traditional ethical boundaries and demand further refinements and clarity of moral theory, ethics itself has not changed nor has its fundamental precepts.

Regardless of how one views this debate, it is clear that the field of computer ethics advances our understanding of key values that lie at the intersection of computer technology and human lives. However, there are some limitations to the field's contributions to HCI. For one, computer ethics often remains too divorced from technical implementations. How exactly, for example, can HCI professionals build interfaces that enhance trust within a community of users? How exactly do we address the value problems that arise through invisible computing? In addition, computer ethics has focused too often on a single value at a time. Yet, as HCI professionals, we commonly wrestle with design trade-offs between competing moral values. Collaborations between computer ethicists and HCI professionals would be a fruitful way to address these limitations.

Social Informatics

In the second section, we reviewed embodied, exogenous, and interactional positions on how values become implicated in technology. Recall that the embodied position holds that designers inscribe their own intentions and values into the technology; once developed and deployed, the resulting technology is said to determine specific kinds of human behavior. As noted, many researchers find such a position highly untenable and instead emphasize the social context in which information systems are used. Recall, for example, Orlikowski's (2000) study, reviewed earlier, where she investigated the effects when Lotus Development Corporation's *Notes* was introduced into a large corporation. She found that, at least in part, the incentive systems in the corporation did more to influence how and whether *Notes* was used than the mere capability of the software itself.

Over the last 25 years, this emphasis on the social context of information technologies has been the subject of systematic research (Attewell, 1987; Borgman, 2000; Iacono & Kling, 1987; Kiesler, 1997; King, 1983; Kling, 1980; Orlikowski, 1993; Poltrock & Grudin, 1994). The research has appeared under many labels, including social analysis of computing, social impacts of computing, information systems, sociotechnical systems, and behavioral information systems. More recently, this overarching enterprise has begun to coalesce within a new field called "social informatics" (Kling, Rosenbaum, & Hert, 1998; Kling & Star, 1998; Sawyer & Rosenbaum, 2000). As defined by participants at the1997 National Science Foundation-sponsored workshop on this topic, *social informatics* is the interdisciplinary study of the design, uses, and consequences of information technologies that takes into account their interactions with institutional and cultural contexts.

To illustrate the value of social informatics, Kling (1999) contrasted the design and functioning of two electronic journals: *Electronic Transactions of Artificial Intelligence* (ETAI) and *Electronic Journal of Cognitive and Brains Sciences* (EJCBS). Both journals envision attracting high-quality papers. Both journals also have implemented technology that works effectively. However, Kling argued that differences in their sociotechnical designs lead ETAI to prosper and EJCBS to wane. For example, articles submitted to ETAI are reviewed in a two-phase process. In the first phase, the article is open to public online discussion for a period of three months. Based on the resulting discussions, authors have an opportunity to revise their papers. In the second phase, the article undergoes a quick, confidential peer review.

In contrast, articles submitted to EJCBS are evaluated by their general online readership. Articles that receive a minimum score are then transferred to an archive of accepted papers. EJCBS has been designed as much as possible as an autonomous system that would run on its own after it was programmed. It removes editorial attention from the publishing process, and instead relies on a readers' plebiscite. However, such a design, according to Kling, misconstrues the social context of successful academic journal publishing, one that requires a lively group of authors and readers and attention from senior scientists in the field.

Work in social informatics has been successful in providing sociotechnical analyses of deployed technologies. Yet, in terms of its contributions to HCI, Johnson (2001), for example, wrote, "One aspect that still confounds me is how to reconcile the basic premise of social informatics—that it is critical to gain knowledge of the social practices and values of the intended users—with the basic work of system developers. How, if at all, can programmers practice and apply social informatics?" (p. 18). Granted, Kling (1999) recognized the importance, for example, of "workplace ethnography, focus groups, user participation in design teams, and participatory design strategies" (Social-technical Systems, para. 4). Then he said, "To discuss it [these methods] in detail here would lead us away from our focus on the structural elements of a socio-technical analysis" (Social-technical Systems, para. 4). Thus, it appears that social informatics can move in at least two directions. One direction leads to developing the sociotechnical analyses and to viewing this work as complementing work in design (and other areas, such as computer ethics). Another direction leads to an expansion of social informatics such that it fundamentally embraces design (and other areas) into its theoretical framework.

Computer-Supported Cooperative Work

Although social informatics has emphasized the sociotechnical analyses of deployed technologies, the field of computer-supported co-operative work (CSCW) has, for some time, focused on the design of new technologies to help people collaborate effectively in the workplace (Galegher, Kraut, & Egido, 1990; Grief, 1988; Grudin, 1988). "Groupware" is the name often used for software that seeks to facilitate CSCW goals. The history of CSCW goes back to early work in various research laboratories, like Xerox PARC, EuroPARC, Bell Lab, and IBM. There, computer professionals working within relatively small groups themselves and sometimes remotely sought to improve their collaborations.

Typically, the values considered in CSCW designs have been closely tied to group activities and workplace issues. Cooperation has been, of course, an overarching value. In addition, the field has paid attention to such values as privacy, autonomy, ownership, commitment, security, and trust. Isaacs, Tang, and Morris (1996), for example, designed a system to support informal interactions in the workplace, using the piazza (the plaza) as the metaphor. The application provided means for workers to opt out of piazza interactions, thus protecting workers' privacy and autonomy. Olson and Teasley (1996) reported on the planning, implementation, and use of groupware tools over the course of a year in a real group with remote members. One of their key findings was that "social responsibility and commitment appeared diminished or missing when people did not meet face-to-face" (Olson & Teasley, 1996, p. 425). Dewan and Shen (1998) discuss an access-control model that accounts for joint ownership of shared objects, different ownership rights for different types of users, and the delegation of access rights (security). Hudson and Smith (1996) sought after methods that allow workers to share video information about themselves to their colleagues while providing protections for privacy. One solution was to shadow images on the video screen so that colleagues can tell that you are in your office but cannot tell what you are doing. Fuchs (1999) developed a notification service for awareness information that also addresses potential conflicts between awareness and privacy. Van House, Butler, and Schiff (1998) examined how in a workplace with environmental planning data sets that trust is created and assessed, and "how changes in technology interact with those practices of trust" (p. 335).

CSCW has traditionally focused on the workplace, as its name implies. Yet if the recent Conference Proceedings of CSCW is any indication, then the field is quickly expanding to include nonworkplace settings. For example, in the CSCW 1996 Proceedings, about 6% of the papers involved a nonworkplace setting. In 1998, it was 13%; in 2000, 31%; in 2002, 36%; and in 2004, 25%. The CSCW 2000 conference co-organizers noted:

> The fact that these [work]place technologies are available not only in working settings but also in homes means that the focus of our attention [in the CSCW community] has broadened to encompass a much wider range of activities than we could have imagined when the CSCW conference series began in 1986. (Dourish & Kiesler, 2000, p. v.)

Correspondingly, the range of values that the CSCW field has begun to analyze has also started to expand. For example, Mynatt et al. (1999) delineated the values of safety, civility, warmth, and friendship that are fostered by SeniorNet, an organization that brings seniors together through computer networking technologies. O'Neill and Gomez (1998) described a project that links middle and high school science students with working scientists as mentors for the students' science projects. The researchers "illustrate the unique dynamics of these relationships, consider their technical and social demands, and discuss the potential for CSCW systems to help sustain long-term help relationships by better accommodating their needs" (O'Neill & Gomez, 1998, p. 325). McCarthy and Anagnost (1998) explored the social ramification of a group preference agent for music in a shared environment—a fitness center. By technical means, they thus seek to democratize the music selection process. Palen, Salzman, and Youngs (2000) tracked 19 new cell mobile phone users for six weeks. In their discussion, they called attention to issues of unfairness that arose in the context of use.

It remains unclear how much more the field of CSCW can expand to include nonworkplace settings before its very name (that has "work" in its title) becomes an historical artifact rather than a description of its current activity. That is a quibble with nomenclature. The direction of the field seems intellectually vibrant and exciting. Thus, as CSCW continues to expand into a broader range of human activity, it will increasingly encounter and thereby take hold of a broader range of human values with ethical import.

Thus, a substantive question emerges: How should we understand the epistemic standing of moral values within a CSCW

framework? One way of currently reading the field is that moral values are simply relative to culture (and to the culture of any particular work group). Harrison and Dourish (1996), for example, argued that "privacy is relative, not a set of psychological primitives" (p. 71). As such, these values should be respected only if the workers themselves think the values are important. Yet, as the field continues to broaden, the tensions embedded in this perspective will continue to become a cause of concern. For example, Greenbaum (1996) argued that CSCW needed to consider the political dimensions of labor and not assume that work is co-operative. Suchman (1994), too, emphasized the power hierarchies within social organizational groups. In such situations, which presumably arise within every organization, at least to some extent, values often will conflict. Perhaps management (which holds the power) seeks efficiency over privacy, whereas workers seek the converse. What does a designer do? Thus, once CSCW analyses move beyond largely homogeneous groups and into organizational structures, then potentially a principled position on the moral standing of human values will be required.

Participatory Design

In Norway, in the early 1970s, there was a general consensus that computer systems should not deskill workers, but enhance skill, protect crafts, and foster meaningful work. At that time, strong labor unions also helped enact into law a national codetermination agreement. This agreement entitled workers along with management to determine which technologies are introduced into the workplace (Kuhn & Winograd, 1996). Thus, emerging from this social structure was a new approach to system design—participatory design—that fundamentally sought to integrate workers' knowledge and a sense of work practice into the system design process (Bannon, 1991; Bjerknes & Bratteteig, 1995; Bødker, 1990; Kensing, Simonsen, & Bødker, 1998, 2004; Ehn, 1989; Floyd, Mehl, Reisin, Schmidt, & Wolf, 1989; Greenbaum & Kyng, 1991; Kyng & Mathiassen, 1997).

In light of value considerations, then, participatory design has embedded within it a commitment to democratization of the workplace and human welfare. It also has what can be viewed as virtue-based moral commitments, as it seeks to account for meaningful activity in everyday lives.

At least five important methods have emerged from, or have been elaborated by, the field of participatory design. (a) Identifying stakeholders—Toward achieving designs that work, it is often necessary to identify the people they directly and indirectly effect (cf., Korpela et al., 1998). (b) Workplace ethnography—Ethnographies document the practices and beliefs of a group from the group's perspective. Methods include analyzing documents and artifacts in the group's environment, participant observation, field observations, surveys, and formal and informal interviews (cf., Kensing, Simonsen, & Bødker, 1998, 2004; Mackay & Fayard, 1999). (c) Future workshops—A future workshop is a method to uncover common problems in the workplace and to solve them. This method is divided into three phases. As described by Kensing and Masden (1991), "The Critique phase is designed to draw out specific issues about current work practice; the Fantasy phase allows participants the freedom to imagine 'what if' the workplace could be different; and the Implementation phase focuses on what resources would be needed to make realistic changes" (p. 157). (d) User participation in design teams—Four ways users participate in design teams can be characterized (Kuhn & Winograd, 1996): directness of interaction with designers, length of involvement in the design process, scope of the participation, and the degree of control over the design decisions. User participation has been central to participatory design projects conducted in Europe and North America. In addition, Korpela et al. (1998) wrote that their seven-year experience in Nigeria suggests that user participation is also a must in developing countries. (e) Mock ups and prototypes—Both mock ups and prototypes create physical representations of technological designs. A mock up looks roughly like the artifact it represents, but completely lacks the artifact's functionality. Mock ups usually occur very early in the design process. As noted by Ehn and Kyng (1991), mock ups encourage hands-on experience and are understandable to the end user, cheap to build, and fun to work with. In turn, prototypes incrementally embed functionality into the artifact through successive iterations. Both methods help end users envision the potential for the proposed technology and the resulting changes in work practice (Greenbaum & Kyng, 1991).

Some HCI practice in the United States has followed closely in the Scandinavian style of participatory design. However, what has been embraced even more commonly is *pragmatic participatory design*, a term we use to refer to the above rich constellation of methods and design techniques but largely stripped of their moral commitments.

Two reasons may help account for why the HCI community in the United States has been resistant to embrace the moral commitments of participatory design, while embracing many of its methods. One reason is that although the United States is politically committed to the value of participatory democracy, its capitalistic business culture is not. Thus, the moral values that lie at the core of participatory design (participation, democracy, and moral personhood) run counter to values in the U.S. workplace. A second reason can be viewed in light of the cultural homogeneity in the Scandinavian countries. Compared with the United States, for example, these countries historically have been more homogeneous in terms of race, ethnicity, and religion, and thus have encountered fewer opportunities for corresponding prejudices and hostilities. Thus, when applied in more diverse contexts, participatory design has too little to say when divisive constituencies argue on the basis of narrowly conceived self-interests and hostile prejudices. After all, at least in theory, participatory design values each participant's voice, even those that appear uncaring and unjust. This problem has been of concern within the field (Gross, Parker, & Elliott, 1997; Muller, 1997).

Value Sensitive Design

Given the limitations of the other approaches in integrating ethics and sociotechnical analyses with actual design, another approach has recently emerged called "value sensitive design" (Friedman, 1997a, 2004; Friedman, Kahn, & Borning, in press). This approach seeks to design technology that accounts for human values in a principled and comprehensive manner throughout the design process. Value sensitive design is primarily concerned with values that center on human well-being, dignity, justice, welfare, and rights. This approach is principled in that

it maintains that such values have moral epistemic standing independent of whether a particular person or group upholds such values. At the same time, value sensitive design maintains that how such values play out in a particular culture at a particular point in time can vary, sometimes considerably.

Value sensitive design articulates an interactional position for how values become implicated in technological designs. From this position, Friedman and Nissenbaum (1996) analyzed bias in computer systems. Cranor and Resnick (2000) analyzed anonymity in ecommerce. Agre and Mailloux (1997) analyzed privacy in computerized vehicle-highway systems. Thomas (1997) analyzed universal access within a communications company. Ackerman and Cranor (1999) analyzed interface components to safeguard users' privacy on the Internet. Cummings (in press) analyzed the impact of interface design of decision support systems on human error and accountability. Friedman, Kahn, Hagman, Severson, and Gill (in press) investigated people's views and values about privacy in public given the widespread use of cameras and display technology. Borning, Friedman, Davis, and Lin (2005) investigated the design of indicators for large-scale urban simulation to support public deliberation around urban planning. Kahn, Freier, Friedman, Severson, and Feldman (2004) examined the social and moral implications of human-robotic interaction. Hagman, Hendrickson, and Whitty (2003) examined informed consent with machine-scannable drivers' licenses. Abowd and Jacobs (2001) called attention to how the design of advanced sensing technologies within and outside the home can impinge on the individual's right (as established by the Fourth Amendment) to be protected against illegal search and seizure by the government. Shneiderman and Rose (1997) proposed social impact statements for information systems. (For related approaches, see Flanagan, Howe, & Nissenbaum, 2005, who examine designer values; Sengers, Boehner, David, & Kaye, 2005, who embed value analyses in their work on reflective design.)

Methodologically, at the core of value sensitive design lies an iterative process that integrates conceptual, empirical, and technical investigations. Conceptual investigations involve philosophically informed analyses of the central constructs and issues under investigation. Questions include how are values supported or diminished by particular technological designs? Who is affected? And how should we engage in trade-offs among competing values in the design, implementation, and use of information systems? Empirical investigations involve social-scientific research on the understandings, contexts, and experiences of the people affected by the technological designs. Technical investigations involve analyzing current technical mechanisms and designs to assess how well they support particular values, and conversely, identifying values, and then identifying and/or developing technical mechanisms and designs that can support those values. As mentioned, these investigations are iterative and integrative. For example, results from the empirical investigations may reveal values initially overlooked in the conceptual investigations or help to prioritize competing values in the design trade-offs between technical mechanisms and values considerations. To illustrate this methodology, consider a recent project by Friedman, Felten, and their colleagues as they sought to understand how to design web-based interactions to respect informed consent (Friedman, Howe, & Felten, 2002; Millett, Friedman, & Felten, 2001). They began their project with a conceptual investigation of informed consent itself. What is it? How can it be garnered in diverse online interactions in general, and in web-based interactions in particular? To validate and refine their resulting conceptual analysis, and initiate their technical investigation, they conducted a retrospective analysis of existing technology. Namely, they examined how the cookie and web-browser technology embedded in Netscape Navigator and Internet Explorer changed—with respect to informed consent—over a five-year period, beginning in 1995. (These results are summarized elsewhere in this chapter.) Then, the design work began. Their design improvements are being implemented in the Mozilla browser (the open-source code for Netscape Navigator) and undergoing empirical investigations (usability studies and formative evaluation), which will then reshape the initial technical and conceptual work.

The National Science Foundation recently sponsored two workshops to help shape a research agenda for Value Sensitive Design. The recommendations from the workshops' final reports (Friedman, 1999; Friedman & Borning, 2001) have focused on the need for (a) theoretical and conceptual analyses that study not only particular values in the online context, but also complexities that arise when tradeoffs among competing values are required in a design; (b) translations of well-analyzed values into technical implementations; (c) proof-of concept projects in which multidisciplinary teams apply Value Sensitive Design to a particular technology, domain, or design problem; (d) contextual analyses that investigate the impact of different stakeholders who influence the design and use of a technology, and who may have different goals and priorities that, in turn, lead to different value tradeoffs; (e) integrating the methodology into organizational structures and work practices; and (f) criteria and metrics—both qualitative and quantitative—that can guide the design process and assess the success of particular designs.

HUMAN VALUES WITH ETHICAL IMPORT

We review and discuss 12 specific human values with ethical import. Some of these values have garnered individual chapters in this handbook. But, by including these values here, we highlight their ethical status and thereby suggest that they have a distinctive claim on resources in the design process.

There are two caveats. First, not all the values we review are fundamentally distinct from one another. Nonetheless, each value has its own language and conceptualizations within its respective fields, and thus warrants separate treatment here. Second, this list is not comprehensive. Perhaps no list could be, at least within the confines of a chapter. Peacefulness, compassion, love, warmth, creativity, humor, originality, vision, friendship, cooperation, collaboration, purposefulness, devotion, diplomacy, kindness, musicality, harmony—the list of other possible values could get very long very fast. Our particular list comprises many of the traditional values that hinge on the deontological and consequentialist moral orientations reviewed previously: human welfare, ownership and property, privacy, freedom from bias, universal usability, trust, autonomy, informed consent, and accountability. In addition, we have chosen several nontraditional values within the HCI community: identity, calmness, and environmental sustainability. Our goal here is not only to point

to important areas of future inquiry, but also to illustrate how an overarching framework for human values and ethics in design can move one quickly and substantively into new territory.

Human Welfare

Perhaps no value is more directly salient to individuals at large as their own welfare, and the welfare of other human beings. Moreover, societal interest in the moral dimensions of computing often arises when harms occur. The faulty computer technology and interface design of Therac-25 led to the physical suffering and death of cancer patients (Leveson & Turner, 1993). Faulty computer technologies are implicated in nuclear accidents like Chernobyl. Indeed, the risks literature (Neumann, 1995) is full of almost daily examples.

Some people in the HCI community, like Leveson (1991), referred broadly to harms that impact people and objects as harms to physical welfare. But there is merit in demarcating three categories of welfare claims (Friedman, 1997b; Kahn, 1992, 1999; Turiel, 1983). *Physical welfare* appeals to the well-being of individuals' biological selves, which is harmed by injury, sickness, and death. Many of the previous examples—Therac-25 problems, "smart" missiles gone astray, nuclear accidents, and so forth—involve harms of this kind. In many such instances, physical harms occur because of faulty hardware or software. But it is important to recognize that technological designs working correctly can themselves lead to corresponding harms or benefits. For example, by means of the connectivity of the web, stalkers can more easily find their "prey," and pedophiles can more easily find unwitting children to contact and potentially abuse. On the positive side, the mediating quality of the web can enhance anonymity (e.g., in a chat room) and buffer individuals from physical harm. *Material welfare* appeals to physical objects that humans value and human economic interests. It goes hardly without saying that a major driving force behind computer technologies in general has been to enhance the material welfare of humans. In turn, a good deal of effort is spent protecting against material harms that computers can engender, such as damaged data from computer viruses and stolen financial information. Much of the work on usable security is relevant here (Cranor & Garfinkel, 2005; Dourish, Grinter, Delgado de la Flor, & Joseph, 2004). *Psychological welfare* refers to the higher-order emotional states of human beings, including comfort, peace, and mental health. The connectivity offered by the web again provides a clear example of psychological benefits and harms. A benefit accrues when the web enhances friendships; and a harm occurs when the web allows for new forms of betrayal (e.g., the "friend" you thought you found in a chat room turns out to be a Bot).

Ownership and Property

From the anthropological literature, it would appear that all cultures embrace the idea that people can own property, although the form of such conceptions can vary considerably (Herskovits, 1952). In perhaps its most stringent form, ownership can be understood as a general right to property, which, in turn, entails a group of specific rights, including the right to possess an object, use it, manage it, derive income from it, and bequeath it (Becker, 1977).

This basic concept of a property right seems simple enough when applied to tangible objects. If you own a table, for example, you can keep it in your house, eat on it, let others eat on it, rent it, and give it away. But current computational technologies blur the boundaries between the tangible and intangible. Herein are difficult questions. Can users, for example, legitimately copy software for their personal use? After all, to do so involves no loss of a physical object from the original seller. Or can a programmer modify part of the code from an operating system and then sell the modified system? Provisional legislative answers in the United States have hinged on nuanced distinctions between copyright, patent, licensing, and trade secrets. For example, unlike a patent, a copyright does not protect ideas, but only the expression of an idea once it is fixed in a tangible medium. The courts have largely granted developers copyrights (but not patents) to their software. But even here, further controversies arise. Let us say that a developer does not copy any code of a competitor's software, but copies its "look and feel"? Is that an infringement? The courts have said it is if other technical means readily exist for implementing a different "look and feel" for the interface (Lipinski & Britz, 2000).

While case law is solidifying for issues that focus on computer software, it is hardly keeping pace with other computational developments that impact ownership and property. Consider, for example, a workplace discussion group. Who gets to decide whether the comments from the online discussion forum are made accessible to the public at large? In other words, do the workers have equal ownership of the compiled contributions? Or is ownership divided by level of participation? Is the owner the organizer of the discussion group, or the president of the company? Can one member delete the contribution from another member (if, for example, he or she finds the contribution offensive)? Consider Goldberg's (2000) Telegarden, an installation that combined robotics and a real-time web camcorder, such that users interacted from a remote distance with a real physical environment (a garden), and planted and watered seeds. Who owns each actual live plant? The user? The owner of the installation? Goldberg himself? Who decides how to respond to an infestation of pests, and whether pesticides will be used? If the garden becomes overcrowded, do the people who physically maintain the site have the right to uproot all the plants that were planted and cared for remotely by others? Or consider whether a cookie on a user's machine belongs to the user or to the website that set the cookie. If the cookie belongs to the user, and the browser does not allow the average user to delete the cookie, then by virtue of a technical mechanism HCI designers have in effect deprived an average user of the capability to exercise one of his or her specific property rights.

Our point is that, through conventions and technical mechanisms, HCI designers shape answers to questions of ownership and property that lie at the forefront of social discourse and legal codification.

Privacy

Privacy refers to a claim, an entitlement, or a right of an individual to determine what information about himself or herself can

be communicated to others (Schoeman, 1984). Historically, a good deal of our privacy protections arose because it was simply too much effort to collect and sort through relevant information about other individuals. But as computer technologies increasingly garner large amounts of information about specific individuals, and increasingly link information databases, our historical protections for privacy are being eroded.

In the HCI community, three general approaches have emerged for privacy protections. One approach informs "people when and what information about them is being captured and to whom the information is being made available" (Bellotti, 1998, p. 70). For example, a video monitor next to a surveillance camera in a convenience store can inform customers of the information being recorded. A second approach allows "people to stipulate what information they project and who can get hold of it" (Bellotti, 1998, p. 70). Tang's (1997) example of an on/off switch for a video conferencing workstation (noted earlier) mapped onto this approach. A third approach applies privacy-enhancing technologies (PETs) that prevent sensitive data from being tagged to a specific individual in the first place. PETs have become "extraordinarily successful" in taxing "the capacities of even the most powerful surveillance institutions" (Phillips, 1998, p. 245), and thus have provoked sharp conflicts in attempts to disseminate them. Indeed, Agre (1998) argued that PETs "disrupt the conventional pessimistic association between technology and social control. No longer are privacy advocates in the position of resisting technology as such" (p. 4).

As we move toward an era of ubiquitous computing, in which embedded computation in our everyday physical objects are linked, the amount of information known about us will increase by many orders of magnitude. Thus, here, as in other domains of applications, we in the HCI community face difficult questions. How do we inform users about the risks to privacy they incur by using various information systems? How do we inform users about the technical protections for privacy available to them? How do we help users to understand the risks from aggregated versus individual data? How much protection is afforded by anonymity? What should the default on systems be?—toward greater privacy protection or toward greater access to information? Given the centrality of these and related questions, privacy has become by far the most visible and well-studied value in HCI (Boyle, Edwards, & Greenberg, 2000; Cohen, Cash, & Muller, 2000; Consolvo, Smith, Matthews, LaMarca, Tabert, & Powledge, 2005; Friedman, Kahn, Hagman, Severson, & Gill, in press; Godefroid, Herbsleb, Jagadeesan, & Li, 2000; Good & Krekelberg, 2003; Iachello & Abowd, 2005; Jancke, Venolia, Grudin, Cadiz, & Gupta, 2001; Jiang, Hong, & Landay, 2002; Milewski & Smith, 2000; Neustaedter & Greenberg, 2003; Palen & Dourish, 2003; Patil & Lai, 2005; Shoemaker & Inkpen, 2001; Tan & Czerwinski, 2003).

Freedom from Bias

Bias refers to systematic unfairness perpetrated on individuals or groups. Three forms of bias in computer systems have been identified (Friedman & Nissenbaum, 1996): preexisting social bias, technical bias, and emergent social bias.

Preexisting social bias has its roots in social institutions, practices, and attitudes. It occurs when computer systems embody biases that exist independently of, and usually prior to, the creation of the software. For instance, the work by Nass and his colleagues (Nass & Brave, 2005; Reeves & Nass, 1996) has shown that people respond to a computer's "gender" along stereotypical lines. Male voice interfaces are rated more competent and persuasive, and more knowledgeable about technical subjects. Female voice interfaces are viewed as more knowledgeable about topics such as love and relationships. As Nass points out, as a designer it is all too easy to inadvertently build on these preexisting social biases when building interfaces. Thus, to minimize preexisting bias designers must not only scrutinize the design specifications, but must couple this scrutiny with a good understanding of relevant biases out in the world. In addition, it can prove useful to identify potential user populations that might otherwise be overlooked and include representative individuals in the field-test groups. Rapid prototyping, formative evaluation, and field testing with such well-conceived populations of users can be an effective means to detect unintentional biases throughout the design process.

Technical bias occurs in the resolution of technical design problems. For example, imagine a database for matching organ donors with potential transplant recipients. If certain individuals retrieved and displayed on initial screens are favored systematically for a match over individuals displayed on later screens, technical bias occurs. Technical bias also originates from attempts to make human constructs such as discourse, judgments, or intuitions amenable to computers. For example, consider a legal expert system that advises defendants on whether or not to plea bargain by assuming that law can be spelled out in an unambiguous manner and is not subject to human interpretations in context. Toward minimizing technical bias, designers often need to look beyond the features internal to a system and envision the design, algorithms, and interfaces in use.

Emergent social bias emerges in the context of the computer system's use, often when societal knowledge or cultural values change, or the system is used with a different population. Toward minimizing emergent bias, designers need to plan for not only a system's intended contexts of use, but also its potentially emergent contexts. Yet, given limited resources, such a proposal cannot be pursued in an unbounded manner. Thus, three practical suggestions are as follows. First, designers should reasonably anticipate probable contexts of use and design for these systems. Second, where it is not possible to design for extended contexts of use, designers should communicate to users the contextual constraints. As with other media, we may need to develop conventions for communicating the perspectives and audience(s) assumed in the design. Third, system designers and administrators can take responsible action if bias emerges with changes in context. The National Resident Medical Match Program offers a good example. Although the original design of the admissions algorithm did not deal well with the changing social conditions (when significant numbers of dual-career couples participated in the match), those responsible for maintaining the system responded conscientiously to this societal change and modified the system's algorithm to place couples more fairly (Roth, 1990).

Universal Usability

Universal usability refers to making all people successful users of information technology. Or, if this requirement of "all people"

is too stringent, it can be reframed as "all people who so desire," or in some other way. Shneiderman (2000), for example, said that universal usability means "having more than 90 percent of all households as successful users of information and communication services at least once a week" (p. 85). In many respects, universal usability comprises a special case of freedom from bias, one that focuses on usability as a means to systematically prevent unfair access to information systems.

Three challenges are often addressed by proponents of universal usability: (a) *technological variety*—supporting a broad range of hardware, software, and network access; (b) *user diversity*—accommodating users with differences in, say, skills, knowledge, age, gender, disabilities, literacy, culture, and income; and (c) *gaps in user knowledge*—bridging the gap between what users know and what they need to know (Shneiderman, 2000). Toward addressing these challenges, current work includes Aberg and Shahmehri (2001); Beard and Korn (2001); Cooper and Rejmer (2001); Jacko, Dixon, Rosa, Scott, and Pappas (1999); Lazar, Dudley-Sponaugle, and Greenidge (2004); Mankoff, Fait, and Tran (2005); Oviatt (1999), and Shneiderman (2003). Stephanidis (2001) provided an anthology of current concepts, methods, and tools.

Universal usability, of course, is not necessarily always a moral good insofar as it depends on what is being used or accessed. Virtually no one would suggest, for example, that there is a moral imperative to provide universal access to reruns of "I Love Lucy." Nonetheless, universal usability often is a moral good. Hert and her colleagues (Hert, Liddy, Shneiderman, & Marchionnini, 2003), for example, showed that if all U.S. citizens have the right to access federal statistics, and if those statistics are only available in an online format, then to be able to exercise their right, the online federal statistics system must be usable by all U.S. citizens. Moreover, universal access with ethical import often provides increased value to a company. For example, based on his case study within a large communications corporation (NYNEX), Thomas (1997) showed how making communications systems more accessible leads to three direct corporate benefits: (a) increasing access makes a communications device more pervasive in social life, and thereby more valuable for everyone; (b) increasing access increases market share; and (c) increasing access "forces technologists, developers, marketers, and executives to think 'out of the box'" (p. 271).

Trust

People sometimes use the term *trust* broadly to include expectations of natural phenomena or machine performance. It is in this sense that people trust that "the sun will rise tomorrow" or that "brakes will stop a car." Indeed, the Computer Science and Telecommunications Board in their thoughtful publication *Trust in Cyberspace* (Schneider, 1999) adopted the terms *trust* and *trustworthy* to describe systems that perform as expected along the dimensions of correctness, security, reliability, safety, and survivability. However, equating the term *trust* with expectations for machine performance (or physical phenomena) misconstrues fundamental characteristics of this value. Specifically, trust is said to exist between people who can experience good will, extend good will toward others, feel vulnerable, and experience betrayal (Baier, 1986; Friedman, Kahn, & Howe, 2000; Kahn & Turiel, 1988). Moreover, on the societal level, trust enhances our social capital (Putnam, Leonardi, & Nanetti, 1993).

In their analysis of trust online, Friedman et al. (2000) suggested that it is important for designers to distinguish two overarching contexts: e-commerce and interpersonal relationships. In e-commerce, for example, certain characteristics of the technology—such as those that concern security, anonymity, accountability, and performance history—can make it difficult for consumers to assess their possible financial harms and the potential good will of the company. One solution that has emerged has been a form of insurance (usually through the credit card companies) that limits a person's financial liability. Trust in e-commerce has received a good deal of attention in the last several years in the HCI and larger community (Dieberger, Höök, Svensson, & Lonnqvist, 2001; Egger, 2000; Fogg & Tseng, 1999; Greenspan, Goldberg, Weimer, & Basso, 2000; Jones, Wilkens, Morris, & Masera, 2000; Olson & Olson, 2000; Riegelsberger, Sasse, & McCarthy, 2003).

In online interpersonal interactions, violations of trust make us most vulnerable psychologically—for example, through hurt feelings or embarrassment. Rocco (1998) suggested that interpersonal trust online succeeds best when preceded by face-to-face interaction. In response, Zheng, Veinott, Bos, Olson, and Olson (2002) suggested that when users do not meet physically but engage in online chat to get to know one another they can establish the same kinds of interpersonal trust that are established in face-to-face interactions (cf. Bos, Olson, Gergle, Olson, & Wright, 2002). Moreover, geography matters for remote collaborations; Bradner and Mark (2002) showed that people, in effect, "trust" less those individuals with whom they interact remotely when they believe those individuals to be in a distant city rather than a nearby one. Interestingly, whereas the characteristic of anonymity works against establishing financial trust in e-commerce, it is double-edged in interpersonal relationships. On the negative side, online anonymity can limit the depth of interpersonal interactions in so far as we engage in a singular means of expression (written). On the positive side, online anonymity opens up new interpersonal opportunities. For example, a gay teenager in an intolerant family and community might rely on the anonymous characteristics of the web to find and interact with like-minded peers. Thus, toward enhancing interpersonal trust online, we need to build tools that allow users to have easy and refined control about what personal information is made known to others.

Autonomy

People decide, plan, and act in ways that they believe will help them to achieve their goals. In this sense people value their autonomy. It might appear relatively easy for HCI designers to support user autonomy. The idea would be to, whenever possible, provide users with the greatest possible control over the technology. However, the task is harder than that. After all, most users of a word processor have little interest, say, in controlling how the editor executes a search-and-replace operation or embeds formatting commands. Similarly, Höök (1998) suggested users may accept adaptive hypermedia as a means to address in-

formation overload. Or, as demonstrated by Barkhuus and Dey (2003) in a study of context-aware computing with a mobile phone device, users may be willing to relinquish some control in order to achieve gains in usefulness. At the same time, users may be uncomfortable with fully automatic systems. For example, in a study investigating the viability of automatic location-disclosure messages to support social awareness (Iachello, Smith, Consolvo, Abowd, Hughes, Howard, Potter, Scott, Sohn, Hightower, & LaMarca, 2005) participants neither used nor wanted the automatic-message feature. In other words, autonomy is protected when users are given control over the right things at the right time. The hard work arises in deciding what those features are and when those conditions occur.

Toward this end, Friedman and Nissenbaum (1997) identified four aspects of systems that can promote or undermine user autonomy. The first involves system capability. Recall Isaacs, Tang, and Morris's (1996) design of a system to support informal interactions in the workplace, using a piazza as the metaphor. Their application provided a means for workers to opt out of piazza interactions, and thus supported worker autonomy. The second involves system complexity. For example, as previously noted, features of software programs can proliferate at the expense of usability, and user autonomy suffers. The third involves misrepresentation of the system. For example, hyperbolic advertising claims can lead users to develop inaccurate expectations of the system and thereby frustrate users' goals. The fourth involves system fluidity. Over time, user's goals often change. Thus, systems need to provide ready mechanisms for users to review and adapt their systems.

Many systems depend on categorization, and it is here that Suchman (1994) worried that organizations can run roughshod over user autonomy. For, according to Suchman, whoever determines the categories—and how those categories can be used—imputes their own personal values into the system and has power over the user. In response, Winograd (1994) pointed to the frequent need for socially coordinated activity through which groups of people seek to share information and technology. In such activity, Winograd argued, we need some degree of standardization wherein designers impose categories. That is their job. The key, according to Winograd, is to cultivate regularized activity without becoming oppressive. In turn, Malone (1994) responded to Suchman by arguing that not only are categories often useful, but to some extent they are necessary given the structure of human cognition. At the same time, Malone suggests that no category system is complete and that designs need to be adaptable, often by their users.

Informed Consent

Informed consent provides a critical protection for privacy, and supports other human values such as trust and autonomy. Yet, currently, there is a mismatch between industry practice and the public's interest. According to a recent report from the Federal Trade Commission (2000), for example, 59% of websites that collect personal identifying information neither inform Internet users that they are collecting such information, nor seek the user's consent. Yet, according to a Harris Poll (2000), 88% of users want websites to garner their consent in such situations. The Federal Trade Commission (2000, p. iv) hoped that industry would continue to make progress on this problem, in conjunction with its proposed legislation. In turn, the HCI community has been recognizing the need to understand better what constitutes informed consent, and to realize it in online interactions.

Friedman and her colleagues (Friedman, Lin, & Miller, 2005; Millet, Friedman, & Felten, 2001) offered an analysis of what constitutes informed consent, and show how both words—"informed" and "consent"—have import (Faden & Beauchamp, 1986; The Belmont Report, 1978). The idea of "informed" encompasses disclosure and comprehension. Disclosure refers to providing accurate information about the benefits and harms that might reasonably be expected from the action under consideration. Comprehension refers to the individual's accurate interpretation of what is being disclosed. In turn, the idea of "consent" encompasses voluntariness, competence, and agreement. Voluntariness refers to assurance that the action is not controlled or coerced, and that an individual could reasonably resist participation should he or she wish to. Competence refers to possessing the mental, emotional, and physical capabilities needed to give informed consent. Agreement refers to a reasonably clear opportunity to accept or decline to participate. Based on this account, Friedman, Millett, and Felten (2000) offered general design principles for informed consent online, namely (a) deciding whether the capability is exempt from informed consent; (b) being particularly careful when invoking the sanction of implicit consent; (c) having default settings err on the side of preserving informed consent; (d) putting the user in control of the "nuisance factor;" (e) avoiding technical jargon; (f) providing the user with choices in terms of potential effects rather than in terms of technical mechanisms; (g) field testing to help ensure adequate comprehension and opportunities for agreement; and (h) designing proactively.

In conjunction with this model for informed consent online, Millett et al. (2001) examined how cookie technology and web browser designs have responded to concerns about informed consent. Specifically, they documented relevant design changes in Netscape Navigator and Internet Explorer over a five-year period, starting in 1995. Their retrospective examination led them to conclude that while cookie technology has improved over time regarding informed consent, some startling problems remained. They specified six problems and offered design remedies. Further work by Friedman, Howe, and Felten (2002) carried this work forward by redesigning several aspects of the Mozilla Browser. For example, no mechanism existed for users to become aware of when a cookie was set without requiring the user to stop the current task. The Mozilla redesign includes a sidebar that watches for cookies and displays a notice in the user's periphery whenever a cookie is set. The display includes information about the domain and third party status of the cookie; the information remains ready-to-hand-over whenever the user chooses to act upon it.

Accountability

Medical expert systems. Automated pilots. Loan approval software. Computer-guided missiles. Increasingly, computers participate in decisions that affect human lives. In cases of com-

puter failure, there is a common response to blame the computer—"It's the computer's fault." Indeed, Friedman and Millett (1995) found that 83% of undergraduate computer science majors she interviewed attributed aspects of agency—either decision-making and/or intentions—to computers. In addition, 21% of the students consistently held computers morally responsible for error. Yet, if we accept that humans, not computational systems, are capable of being moral agents, such blame is fundamentally misplaced (Searle, 1980).

How can HCI designers minimize this tendency of users to attribute blame to computational systems? Suggestions have been offered based on minimizing two types of distortions (Friedman & Kahn, 1992). In the first type of distortion, the computational system diminishes or undermines the human user's sense of his or her moral agency. In such systems, human users are placed into largely mechanical roles, either mentally or physically, and frequently have little understanding of the larger purpose or meaning of their individual actions. To the extent that humans experience a diminished sense of agency, human dignity is eroded and individuals may consider themselves largely unaccountable for the consequences of their computer use. Conversely, in the second type of distortion, the computational system masquerades as an agent by projecting intentions, desires, and volition. Strikingly, even when computer interfaces only minimally mimic human agency, people appear predisposed—at least in certain regards—to treat computers as social agents. For example, as Nass and his colleagues have shown, people respond to multiple voices from a single computer as though they were separate entities, respond to a computer's "gender" along stereotypical lines, are less likely to criticize a computer directly (e.g., if the computer itself asks for an evaluation) than to criticize the computer to a third party (a different computer or human), and respond to computer flattery (Nass & Brave, 2005; Reeves & Nass, 1996). Thus, to the extent that humans inappropriately attribute agency to computational systems, humans may well consider the computational systems, at least in part, to be morally responsible for the effects of computer-mediated or computer-controlled actions.

Identity

The idea of personal identity embraces two seemingly contradictory ideas. On the one side is the obvious fact that each one of us has many roles. A single person can be, for example, a father, lover, poker player, gourmet cook, computer geek, and animal lover. Indeed, William James says that a person "has many social selves as there are individuals who recognize him" (quoted in Rosenberg, 1997, p. 23). On the other side, virtually all of us feel like we live reasonably coherent lives, and that the person we are today is pretty much the person we were yesterday and last week, if not last year. Thus, identity appears to be multiple and unified, and both aspects are essential to human flourishing. Too far toward multiplicity and we end up schizophrenic; too far toward being unified and there are too few mechanisms for psychological growth, and too little basis for healthy social functioning (e.g., it makes little sense to present the same "persona" to one's boss as one does to one's child).

In terms of HCI, it is important that the field as a whole substantively support both manifestations of identity, and that in our designs we do not swing too far one direction or the other. To date, the networked personal computer has easily supported multiplicity. Thus, the same person can easily communicate with many unrelated groups (e.g., chats, list servers, etc.), easily establish a different identity within many of these groups, or even multiple identities within a single group (Turkle, 1996). This trend is being checked in interesting ways. For example, individuals in some online communities link their comments to their personal homepage ("To find out more about me, click here.") Some chats (e.g., SeniorNet described in Mynatt et al., 1999) require single identities. Ubiquitous location-aware systems (e.g., DigiDress described in Persson, Blom, & Yang, 2005) have been used to facilitate "real" social interaction between previously unacquainted users. Indeed, with the advent of increasingly linked online databases coupled with ubiquitous computing, we believe that the pendulum will soon shift powerfully and pervasively toward the unification of identity. If so, following Bers, Gonzalo-Heydrich, and DeMaso (2001), and Schiano and White (1998), the challenge for HCI designers will be to find ways such that individuals have flexibility to establish and reveal not only an integrated self but also a multiplicity of identities.

Calmness

Weiser and Brown (1997) suggested that in the last 50 years, there have been two great trends in the human relationship with computing: the first was with the mainframe computer and the second is currently with the personal computer. They suggest that the next great trend is toward ubiquitous computing, characterized by deeply embedded computation in the world. In turn, they argue that the "most potentially interesting, challenging, and profound change implied by the ubiquitous computing era is a focus on *calm*. If computers are everywhere, they had better stay out of the way, and that means designing them so that the people being shared by the computers remain serene and in control" (p. 79).

The central design mechanism Weiser and Brown put forward is one in which information remains in the periphery until it is relevant and, only then, is moved to the center of our awareness. One architectural example they provide is of a glass window between offices and hallways. Such a window, they suggest, "extends our periphery by creating a two-way channel for clues about the environment. Whether it is motion of other people down the hall (it's time for lunch; the big meeting is starting) or noticing the same person peeking in for the third time while you are on the phone (they really want to see me; I forgot an appointment), the window connects the person inside to the nearby world" (pp. 81–82). A computational example involves an Internet multicast, a continuous video from another location that provides not so much video conferencing, but "more like a window of awareness" of another location (p. 82). Another example involves a "dangling string," "an eight-foot piece of plastic spaghetti that hangs from a small electric motor mounted in a ceiling. The motor is electronically connected to a nearby Ethernet cable, so that each bit of information that goes past causes a tiny twitch of the motor. A very busy network

causes a madly whirling string with a characteristic noise; a quiet network causes only a small twitch every few seconds" (p. 83).

The idea that ubiquitous computing will need to take up the challenge of preserving calmness in human lives seems on the mark. There are, however, some limitations in the solution Weiser and Brown put forward in terms of designing for the periphery. Perhaps the most notable is that very quickly the periphery itself can become overloaded with information, especially if workplace conventions expect workers to be continuously aware of such information. Imagine, for example, working in an office with a window out onto the hallway, with four Internet multicasts playing in the background, a dangling string in the corner, five other peripheral information streams on each of several pieces of software you are using, and then add in peripheral information streams on one's regular phone, cell phone, personal digital assistant, and any and all other pieces of embedded computation in the office of today (or the future). Then try to get some work done. It will not be easy. Thus, as ubiquitous computing takes hold, it is not surprising that we see an emerging body of work addressing the challenges of information overload and interruptability (Fogarty, Hudson, Atkeson, Avrahami, Forlizzi, Kiesler, Lee, & Yang, 2005; Hudson, Christensen, Kellog, & Erickson, 2002; Voida, Newstetter, & Mynatt, 2002).

An alternative design approach (Kahn & Friedman, in preparation; cf. Farrell, 2001) toward preserving calmness has as its starting point the psychological literature that shows that direct experiences with nature have beneficial effects on people's physical, cognitive, and emotional well-being. For example, Ulrich (1984) found that postoperative recovery improved when patients were assigned to a room with a view of a natural setting (a small stand of deciduous trees) versus a view of a brown brick wall. More generally, studies have shown that even minimal connection with nature—such as looking at a natural landscape—can reduce immediate and long-term stress, reduce sickness of prisoners, and calm patients before and during surgery (see, for reviews, Beck & Katcher, 1996; Kahn, 1999; Ulrich, 1993). Building on this literature, the question becomes in what ways computer technologies can augment the human relationship with the natural world with beneficial effects. One implementation might involve video plasma display "windows" in inside offices that stream in real-time views of a psychologically restorative local nature scene (Friedman, Freier, & Kahn, 2004). Another implementation might involve robotic "pets" as possible companions for the elderly (Beck, Edwards, Kahn, & Friedman, 2004). The point is that while such augmented natural interactions may not be as psychologically beneficial as real nature, it may be more so than no nature, in which case it becomes a plausible area for design applications.

Environmental Sustainability

We pollute air and water, deplete soil, deforest, create toxic wastes, and through human activity are extinguishing over 27,000 species each year (a conservative estimate). Such problems have generated a tremendous amount of attention among the populace, and have slowly been coming under the purview of the computing community (cf. IEEE and ACM: Software Engineering Code of Ethics and Professional Practice, 1998). On the production end, there is concern about the resources used in producing computer technologies, as well as the resulting toxic wastes (e.g., of making computer chips). On the consumption end, there is concern about the resources computer technologies use. The electrical demands, for example, are particularly high, especially when energy sources are scarce.

Another question arises of how we can design systems that foster a healthier and more life-affirming connection with the natural world (cf. Kahn, 2006; Kahn & Kellert, 2002). The more straightforward approach has been simply to harness computer technologies in the service of environmental science, in computer models, for example, of global warming or earth tectonics. Environmental educators have also been using networked computers such that students can share environmental data or narratives with other students (or scientists) in diverse geographical locations. Or consider an approach being taken by Borning and his colleagues (Waddell, Borning, Noth, Freier, Becke, & Ulfarsson, 2003) where they are using a computer simulation ("UrbanSim") of an urban environment to help residents, politicians, and planners visualize the effects of their proposed land-use plans. Another approach might involve designing calming technologies, as previously discussed, that integrate restorative aspects of nature into human lives.

PROFESSIONAL ETHICS

Computer professionals sometimes experience ethical conflicts in the workplace. Consider, for example, the situation in which, by means of his or her specialized knowledge, a computer professional knows of a harm that can result from the implementation of a computer technology. Should he or she inform others of such impeding harm even if it will jeopardize his or her job security? Or consider the situation in which a client asks a software developer to develop an expert system that recommends against loans in certain neighborhoods. Although such a policy (called "redlining") may serve the economic interests of the loan-granting organization, it unfairly discriminates against individuals on the basis of where they live. Should a computer professional deny such a request? Or consider the situation offered by Quesenbery (Molich, 2001):

> You have set up early usability tests of a paper prototype with nurses at a medical facility. The test was difficult to schedule because nurses' time is guarded carefully and marketing carefully guards the relationship with customers. The nursing managers insist on "taking the test" themselves first, and then on being present in the room during the test with the other participants to "be sure they do it right." You believe that the manger's presence will be intimidating to the nurses, altering the results of the test. Do you continue with the usability tests? (p. 218)

Quesenbery offers two possible answers. One answer is yes. She argues that if you turn down this opportunity, another one might not arise; and although you would have to carefully evaluate the test results for bias caused by the managers, valuable data about the prototype would still be generated. The other answer is no. She argues that it is unethical to put nurses into such

a stressful situation in which they are studied in front of their obviously critical managers.

In response to ethical issues that arise within the computing professions, numerous organizations have developed ethical codes of conduct. Codes have been developed, for example, by the ACM, IEEE Computer Society, DPMA, and ICCP. Berleur and Brunnstein (1997) provided a comparison of 30 different codes of ethics. At the end of 1998, the IEEE and the ACM together adopted a revised Software Engineering Code of Ethics and Professional Practice (http://www-cs.etsu.edu/seeri/secode.htm). At the highest level of abstraction (see Gotterbarn (1999) for a review), the code states eight principles wherein software engineers shall (a) act consistently with the public interest; (b) act in a manner that is in the best interests of their client and employer, consistent with the public interest; (c) ensure that their products and related modifications meet the highest professional standards possible; (d) maintain integrity and independence in their professional judgment; (e) subscribe to and promote an ethical approach to the management of software development and maintenance; (f) advance the integrity and reputation of the profession consistent with the public interest; (g) be fair to and supportive of their colleagues; and (h) participate in life-long learning regarding the practice of their profession and promote an ethical approach to the practice of their profession. In turn, each principle is elaborated upon with specific guidelines. For example, under the first principle (to act consistently with the public interest), the code states that software engineers should "disclose to appropriate persons or authorities an actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with the software or related documents" (p. 62).

Before moving forward with this topic, it is important to take hold of a question often discussed in the literature: whether the computer profession is even a profession (Weckert & Adeney, 1997). If it is not, then concerns about professional ethics—special moral requirements above and beyond what are applied to ordinary people—disappear.

What, then, is a profession? According to Bayles (1981), a profession is comprised of members who (a) have extensive training, (b) have an intellectual component such that the professional's primary role is to advise the client about things the client does not know, and (c) provide an important service to society. Bennion (1969) also pointed out that professionals are typically self-employed, and establish a client-agent relationship based on trust. In addition, as Johnson and Miller (1997) wrote: "Being a professional means more than just having a job. The difference is commitment to doing the right thing because you are a member of a group that has taken on responsibility for a domain of social activity—a social function. The group is accountable to society for this domain, and for this reason, professionals must behave in ways that are worthy of public trust" (p. 22).

Based on such criteria, in some respects the computer field constitutes a profession. Computer personnel have intellectual knowledge and extensive training, and they provide an important service to society. Computer personnel are also especially well positioned to understand how new technologies may impact human lives; accordingly, many people argue that they incur the responsibility to communicate such impacts to the public. Also, the media, industry, and academy widely use the term "computer professionals." That said, the computer field certainly does not constitute a canonical profession, like medicine or law. After all, many computer personnel work for businesses rather than as consultants, and create artifacts (e.g., software, hardware, algorithms, interface designs) rather than offer advice. Additionally, computer personnel vary greatly in the amount and type of their training. Some computer personnel are self-taught, without even a high-school diploma. Other computer personnel complete six-month to two-year programs, or a bachelor's, master's, or doctoral program.

If we accept—as most people do—that in at least some important regards the computer field constitutes a profession, then how has the profession understood and conveyed its correlative ethical responsibilities? One approach has been through codes of ethics, as noted previously. Such codes serve multiple purposes (Gotterbarn, 1999; Anderson, Johnson, Gotterbarn, & Perrolle, 1993). They serve to educate computer employees and managers. They help garner the trust of the public. They provide computer employees with a formal document to turn to (and appeal to) when they face conflicts that pit ethical decisions against the economic benefit of their company. They can function as a means of deterrence and discipline. And they can enhance a profession's public standing.

Another approach for advancing the profession's ethical responsibilities involves using hypothetical scenarios to flesh out the meaning of the codes (Parker, 1979; Parker, Swope, & Baker, 1990). For example, Anderson et al. (1993) provided an analysis of nine scenarios that illustrate how the 1992 ACM Code of Ethics bears out in practice. They offered, for example, the hypothetical situation of a consultant, named Diane, who is designing the database-management system for the personnel office of a medium-sized company. The database will store sensitive information, including performance evaluations, medical records, and salaries. In an effort to cut costs, the CEO of the company rejects Diane's suggestions and opts for a less-secure system. Diane remains convinced that a more-secure system is needed. What should she do? Anderson et al. (1993) then pointed to specific passages in the code that pertain to the importance of maintaining privacy and confidentiality. Based on the code, Diane's first obligation is to try to educate the company officials on the ethics of the situation. "If that fails, then Diane needs to consider her contractual obligations . . . on honoring assigned responsibilities" (p. 100).

Another possible approach toward advancing the profession's ethical responsibilities involves that of licensing its members. The model draws from the American Medical Association or the American Bar Association. Both associations can revoke their members' licenses given serious violations of their respective professional organization's code of ethics. But there are good reasons why most professional computing organizations have resisted this move. Computing is an enormously diverse field and it is difficult to formally demarcate all its areas, let alone the relevancy of one area to others. What exactly must an independent website designer know, for example, beyond whatever the designer and employer believe necessary? Does the field really want to exclude people all along its fringes? There has been concern that through licensing the profession might become a closed shop and thereby enhance the status and incomes of those admitted at the expense of those excluded. There has also been concern that licensing could stifle creativity, originality, and excellence (see the discussion by Neumann,

1991). Licensing also runs counter to the history of computing that has emphasized ease of entry, entrepreneurship, and a culture not often found in professional licensing organizations.

Setting aside licensing, and recognizing the value of codes of ethics and analyses of hypothetical dilemmas, what else can the computing profession do to advance its ethical responsibilities? Johnson and Miller (1997) offered two suggestions. One is that corporations could have ombudspersons to whom computing professionals could report their concerns, anonymously if desired. Another is that professional societies (like the ACM) could maintain "hotlines that professionals could call for advice on how to get their concerns addressed" (p. 23). Both suggestions, if implemented, could prevent many problems from escalating into "whistle-blowing" affairs and help integrate ethical considerations into the workplace.

The computing profession is also recognizing the importance of analyzing not only hypothetical scenarios, but also real life events. Neumann (1995) was a pioneer in this regard, publishing ongoing chronicles of the risks that arise through computing. For the more interesting cases, Neumann's approach could be extended. What we have in mind here builds on a leading mountaineering journal that each year publishes brief accounts of the major mountaineering accidents of the previous year. To read these accounts, year in and year out, is to build up a rich repertoire of how mistakes happen in the field, and how they can be avoided. Similarly, it would be possible to provide comprehensive accounts of the major ethical problems that have arisen each year in the computing profession. Moreover, these accounts could be classified based on type of ethical value (e.g., privacy, trust, human welfare, security, or universal usability), user population (e.g., workers, elderly, children, or people in developing nations), and applications (e.g., online communities, information visualization, augmented reality, wearable computing, or groupware). With such a searchable database, computing professionals could gain ready access to real-life ethical case studies directly relevant to their own endeavors. In addition to case studies, Hudson and Bruckman (2005) argued that at times empirical research into users' views and values can provide evidence to inform on answering difficult ethical dilemmas. For example, with online interactions, one key piece of information to understand is whether users consider their online interactions to be public or private.

Most of the ethical issues of HCI professionals are subsumed under those encountered by computer professionals in general. But HCI professionals incur unique responsibilities whenever they involve human subjects. At universities, such studies come under the purview of a Human Subjects Institutional Review Board (an IRB) that seeks to protect the rights and welfare of human subjects. But, in industry, such oversight is usually lacking. Consider the following questions:

- You conduct usability studies with people within your company. The Vice President later asks you which people did well. Do you tell? (Molich, 2001)
- You collect video footage of people engaged in your usability study. Can you show the footage at a conference? (Mackay, 1995) Have you obtained written consent from the subjects? What if the subject initially gives consent but then comes off looking foolish in the video? Have you given the subject a chance to review the video footage and to opt out?
- You obtain informed consent, and then collect keystroke data on employees over months. How do you keep your subjects aware that your data collection is ongoing?
- You collect data from subjects living in an "aware home" (that has ubiquitous computation embedded throughout the living environment). How do you convey to subjects the extent of personal information (e.g., time spent in the bathroom) that can be mined from the resulting database?
- You collect data from employees in your company, and promise to keep identifying information confidential. But if there is only one disabled person in a given department, how can you maintain confidentiality?
- You are involved in a safety-critical situation. You recognize that often usability studies in your company involve only a handful of subjects, which allows for fast prototyping and more quickly bringing products to market. But, in this situation, is a more rigorous psychological study of usability needed, even given that it will require additional resources and time? (Molich, 2001)
- You collect information from an online chat focused on sexual abuse and then publish the information widely, quoting key emotional passages that now become visible on a societal level. Even though you report the passages without identifying names, others can now go to this chat, search on the text, and easily identify the people of interest. Have you adequately protected the privacy of these online "subjects"? How does using subjects garnered from the Internet differ from those garnered from more traditional subject pools (Frankel & Siang, 1999)?

These questions and many other related issues can be addressed by drawing on the rich literature on protecting the rights and welfare of human subjects (*The Belmont Report*, 1978) and its application to the Internet (Frankel & Siang, 1999). Indeed, often HCI professionals may be the only source for giving voice to these ethical considerations.

CONCLUSIONS

During the early periods of computerization, around the 1950s, cyberneticist Norbert Wiener (1953/1985) argued that technology could help make us better human beings and create a more just society. But for it to do so, he argued, we have to take control of the technology. We have to reject the "worshiping [of] the new gadgets which are our own creation as if they were our masters" (p. 678). Similarly, a few decades later, computer scientist Joseph Weizenbaum (1972) wrote:

> What is wrong, I think, is that we have permitted technological metaphors . . . and technique itself to so thoroughly pervade our thought processes that we have finally abdicated to technology the very duty to formulate questions. . . . Where a simple man might ask: "Do we need these things?," technology asks "what electronic wizardry will make them safe? Where a simple man will ask, "Is it good?" technology asks, "Will it work?" (pp. 611–612)

As HCI professionals, we have profound opportunities to shape the designs and implementations of computer technologies from an ethical stance.

In this chapter, we have reviewed varying approaches, projects, and ideas that offer us important ways of bringing human values and ethics into our design practice. Several ideas are worth emphasizing. First, as is well known, it is much easier to design systems right initially than to attempt to retrofit poor systems after they have become entrenched within organizations and other social systems. Thus, it is imperative that we take a proactive stance on human values, ethics, and design. Second, many of the difficult problems in this area require multidisciplinary collaborations. Third, we need to hold out human values with ethical import as a design criterion—along with the traditional criteria of reliability, efficiency, and correctness—by which systems may be judged poor and designers negligent. As with the traditional criteria, we need not require perfection, but commitment.

This *Handbook* has an entire section devoted to issues of diversity. Individual chapters include, for example, designing for gender differences; children; the elderly; internationalization; and people with motor, perceptual, and cognitive impairments. Yet, in our review of the literature, current HCI approaches to human values and ethics do not always fare well when used in diverse contexts. For example, Participatory Design can be particularly effective when a community shares many deeply held sensibilities, such as a commitment to participatory democracy in the workplace and to the idea of meaningful work itself. But this approach is more difficult to apply when divisive constituencies argue on the basis of narrowly conceived self-interests or hostile prejudices. What happens, for example, when a manager values accountability over a worker's privacy, or efficiency over a worker's autonomy? If each value has equal weight, on what basis does a designer move forward with a particular design?

As the field of HCI seeks to bring moral commitments into diverse contexts, it needs a principled moral means to adjudicate competing value claims. It is for this reason that we provided early on a moral philosophical and psychological framework for approaching this problem. We suggested, for example, that when moral conflicts occur between diverse groups, that such variability by itself does not prove or disprove the moral relativist's position. People can believe that a certain act (such as racial discrimination) is moral; but documenting such a belief does not make it so. We also suggested that when moral differences appear in another culture, a question to ask is whether there are victims within that culture. If there are, then it is probably less the case that societies differ on moral ground, and more that some societies (Western societies included) may be involved explicitly in immoral practices. We suggested that close attention must be paid to the level of moral analysis. Namely, definitions of morality that entail abstract characterizations of justice and welfare tend to highlight moral universals, whereas definitions that entail specific behaviors or rigid moral rules tend to highlight moral cross-cultural variation. In our view, both levels of analysis have merit, and a middle ground provides an epistemically sensible and powerful approach in HCI research—one that allows for an analysis of universal moral constructs (such as justice, rights, welfare, and virtue), as well as allowing for the ways in which these constructs play out in a particular culture at a particular point in time. By embracing this moral theoretical framework, we are not saying that all moral problems can be solved. But such problems will be discussed (and argued about) in ways that respect diversity and prevent oppression.

In the future, certain trends—technological and societal—will pose particular challenges in terms of human values, ethics, and design. From our view, three trends stand out. First, computational technologies will increasingly allow for the erosion of personal privacy. Even today, for example, surveillance cameras capture our images in banks and airports, and in many stores, malls, and even streets. In cars, GPS navigation systems not only receive positioning data, but also can broadcast one's position. Businesses can (and sometimes do) monitor workers' electronic communications. Indeed, think of perhaps the last bastion of the private space, the home; and recognize that "aware homes" of the future will have the potential to record virtually every movement an individual makes within his or her home, and to link that data to large networked databases. Thus, protections of individual privacy will need to become an even more central concern to the HCI community. The second trend is that computational technologies will increasingly provide means for government to erode civil liberties. Moreover, the public may unwittingly accept such consequences based on the assumption that these technologies will substantially enhance our nation's physical security. But such an assumption is not always warranted. Imagine, for example, if the government required each of us to obtain biometric national identification cards, and to use these cards for all of our financial transactions, air travel, entry into government buildings, and so forth. Secure? Hardly, for the systems that store this data can be cracked. Indeed, once biometric data is stolen, it is not so easy to get a new face, fingerprint, or DNA. Third, in our increasingly linked communicative-technological infrastructure, the pervasiveness and speed of information can undermine people's psychological well-being and quality of life. How can we respond to this problem of information overload? One solution is that we need, at times, to turn away from our technological devices and thus to check their encroachment into human lives. Another solution is that, in our designs, we need to find ways to increase the ratio of quality over quantity of information. We can also profit by designing "calming" technologies (that go beyond placing information in our peripheral awareness, which itself can become overloaded).

In the last decade, the HCI community has made tremendous progress in integrating human values and ethics into the practice of design. As the field continues to move forward, the challenge remains how to design technology wisely, ethically, to create essential conditions by which humans live and flourish.

ACKNOWLEDGMENTS

We thank Alan Borning and Gaetano Borriello for sustained discussions on human values and computational technologies. Nathan G. Freier, Jessica Miller, and Valerie Wonder assisted with library research. The ideas presented in this paper are, in part, based on work supported by the National Science Foundation under Award Nos. IIS-0000567, IIS-9911185, SES-0096131, and IIS-0325035. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Aberg, J., & Shahmehri, N. (2001). An empirical study of human web assistants: Implications for user support in web information systems. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2000*) (pp. 404–411). New York, NY.

Abowd, G. D., & Jacobs, A. (2001, September/October). The impact of awareness technologies on privacy litigation. *SIGCHI Bulletin, 9.*

Ackerman, M. S., & Cranor, L. (1999). Privacy critics: UI components to safeguard users' privacy. In *Extended Abstracts Conference on Human Factors in Computing Systems* (*CHI 99*) (pp. 258–259). New York: Association for Computing Machinery Press.

Adler, P. S., & Winograd, T. (Eds.). (1992). *Usability: Turning technologies into tools*. Oxford: Oxford University Press.

Agre, P. E. (1998). Introduction. In P. E. Agre & M. Rotenberg (Eds.), *Technology and privacy: The new landscape* (pp. 1–28). Cambridge, MA: The MIT Press.

Agre, P. E., & Mailloux, C. A. (1997). Social choice about privacy: Intelligent vehicle-highway systems in the United States. In B. Friedman (Ed.), *Human values and the design of computer systems* (pp. 289–310). New York: Cambridge University Press.

Agre, P. E., & Rotenberg, M. (1998). *Technology and privacy: The new landscape*. Cambridge, MA: The MIT Press.

Akrich, M. (1992). The de-scription of technical objects. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 205–224). Cambridge, MA: The MIT Press.

Anderson, R. E., Johnson, D. G., Gotterbarn, D., & Perrolle, J. (1993). Using the ACM code of ethics for decision making. *Communications of the ACM, 36*(2), 98–107.

Appadurai, A. (Ed.). (1988). *The social life of things*. New York: Cambridge University Press.

Association for Computing Machinery, Inc. & Institute for Electrical and Electronics Engineers, Inc. (1998) Software engineering code of ethics and professional practice. Retrieved date March 15, 2007, from http://www.acm.org/serving/se/code.htm

Attewell, P. (1987). The deskilling controversy. *Work and Occupation, 14*(3), 323–346.

Baier, A. (1986). Trust and antitrust. *Ethics, 96*, 231–260.

Bannon, L. J. (1991). From human factors to human actors: The role of psychology and human-computer interaction studies in systems design. In J. Greenbaum & M. Kyng (Eds.) *Design at work: Cooperative design of computer systems* (pp. 25–44). Hillsdale, NJ: Lawrence Erlbaum Associates.

Barkhuus, L., & Dey, A. (2003). Is context-aware computing taking control away from the user? Three levels of interactivity examined. *Proceedings of UbiComp 2003* (pp. 149–156). New York, NY.

Bayles, M. D. (1981). *Professional ethics*. Belmont, CA: Wadsworth.

Beard, M., & Korn, P. (2001). What I need is what I get: Downloadable user interfaces vis Jini and Java. *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2001*) (pp. 15–16). New York, NY.

Beck, A., & Katcher, A. (1996). *Between pets and people*. West Lafayette, IN: Purdue University Press.

Beck, A. M., Edwards, N. E., Kahn, P. H., Jr., & Friedman, B. (2004, October). *Robotic pets as perceived companions for older adults.* Paper presented at the 10th International Conference on Human-Animal Interactions, Glasgow, Scotland.

Becker, L. C. (1977). *Property rights: Philosophical foundations*. London: Routledge & Kegan Paul.

Bederson, B. B., Lee, B., Sherman, R. M., Herrnson, P. S., & Niemi, R. G. (2003). Electronic voting system usability issues. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 145–152). New York, NY.

Bellotti, V. (1998). Design for privacy in multimedia computing and communications environments. In P. E. Agre & M. Rotenberg (Eds.), *Technology and privacy: The new landscape* (pp. 63–98). Cambridge, MA: The MIT Press.

Bennett, C. J. (1998). Convergence revisited: Toward a global policy for the protection of personal data? In P. E. Agre & M. Rotenberg (Eds.), *Technology and privacy: The new landscape* (pp. 99–123). Cambridge, MA: The MIT Press.

Bennion, F. A. R. (1969). *Professional ethics*. London: Charles Knight & Co.

Berleur, J., & Brunnstein, K. (Eds.). (1997). *Ethics of computing: Codes, spaces for discussion, and law*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bers, M. U., Gonzalez-Heydrich, J., & DeMaso, D. R. (2001). Identity construction environments: Supporting a virtual therapeutic community of pediatric patients undergoing dialysis. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 2001*) (pp. 380–387). New York, NY.

Bjerknes, G., & Bratteteig, T. (1995). User participation and democracy: A discussion of Scandinavian research on system development. *Scandinavian Journal of Information Systems, 7*(1), 73–97.

Bødker, S. (1990). *Through the interface—A human activity approach to user interface design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Borgman, C. L. (2000). *From Gutenberg to the global information infrastructure: Access to information in the networked world*. Cambridge, MA: The MIT Press.

Borning, A., Friedman, B., Davis, J., & Lin, P. (2005). Informing public deliberation: Value sensitive design of indicators for a large-scale urban simulation. In *Proceedings of ECSCW* (pp. 449–468), Dordrecht, The Netherlands.

Bos, N., Olson, J., Gergle, D., Olson, G., & Wright, Z. (2002). Effects of four computer-mediated communications channels on trust development. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI '02*) (pp. 135–140). Minneapolis, MN.

Boyd, R. A. (1988). How to be a moral realist. In G. Sayre-McCord (Ed.), *Essays on moral realism* (pp. 181–228). Ithaca, NY: Cornell University Press.

Boyle, M., Edwards, C., & Greenberg, S. (2000). The effects of filtered video on awareness and privacy. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*) (pp. 1–10). New York, NY.

Bradner, E., & Mark, G. (2002). Why distance matters: Effects on cooperation, persuasion and deception. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2002*) (pp. 226–235), New York, NY.

Bulliet, R. (1994). Determinism and pre-industrial technology. In M. R. Smith & L. Marx (Eds.), *Does technology drive history? The dilemma of technological determinism* (pp. 201–216). Cambridge, MA: The MIT Press.

Bynum, T. W. (Ed.). (1985). [Entire issue]. *Metaphilosophy, 16*(4).

Campbell, R. L., & Christopher, J. C. (1996). Moral development theory: A critique of its Kantian presuppositions. *Developmental Review, 16*, 1–47.

Cohen, A. L., Cash, D., & Muller, M. J. (2000). Designing to support adversarial collaboration. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*), New York, NY, 31–39.

Cole, M. (1991, April). Discussant to B. Friedman & P. H. Kahn, Jr. (Presenters), *Who is responsible for what? And can what be responsible? The psychological boundaries of moral responsibility.* Paper presented at biennial meeting of the Society for Research in Child Development, Seattle, WA.

Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J., & Powledge, P. (2005). Location disclosure to social relations: Why, when, & what people want to share. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2005*), New York, NY, 81–90.

Cooper, M., & Rejmer, P. (2001). Case study: Localization of an accessibility evaluation. *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2001*) (pp. 141–142), New York, NY.

Cranor, L. F., & Garfinkel, S. (2005). *Security and usability: Designing secure systems that people can use.* Cambridge, MA: O'Reilly.

Cranor, L. F., & Resnick, P. (2000). Protocols for automated negotiations with buyer anonymity and seller reputations. *Netnomics, 2*(1), 1–23.

Cummings, M. L. (in press). The social and ethical impact of decision support interface design. In W. Karwowski (Ed.), *International encyclopedia of ergonomics and human factors* (2nd ed.). New York: Taylor & Francis.

Dewan, P., & Shen, H. (1998). Flexible meta access-control for collaborative applications. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 1998*) (pp. 247–256), New York, NY.

Dieberger, A., Höök, K., Svensson, M., & Lonnqvist, P. (2001). Social navigation research agenda. *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2001*) (pp. 107–108), New York, NY.

Dourish, P., Grinter, R., Delgado de la Flor, J., & Joseph, M. (2004). Security in the wild: User strategies for managing security as an everyday, practical problem. *Personal and Ubiquitous Computing, 8*(6), 391–401.

Dourish, P., & Kiesler, S. (2000). From the papers co-chairs. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*), New York, NY, v.

Dworkin, R. (1978). *Taking rights seriously*. Cambridge, MA: Harvard University Press.

Egger, F. N. (2000). "Trust me, I'm an online vendor": Towards a model of trust for e-commerce system design. *Extended Abstracts of the Conference of Human Factors in Computing Systems* (*CHI 2000*) (pp. 101–102), New York, NY.

Ehn, P. (1989). *Work-oriented design of computer artifacts*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ehn, P., & Kyng, M. (1991). Cardboard computers: Mocking-it-up or hands-on the future. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 169–195). Hillsdale, NJ: Lawrence Erlbaum Associates.

Faden, R., & Beauchamp, T. (1986). *A history and theory of informed consent*. New York: Oxford University Press.

Farrell, S. (2001). Social and informational proxies in a fishtank. *Extended Abstracts of the Conference of Human Factors in Computing Systems* (*CHI 2001*) (pp. 365–366), New York, NY.

Federal Trade Commission. (2000, May). *Privacy online: Fair information practices in the electronic marketplace* (A report to Congress). Washington, DC: Author.

Fishkin, J. S. (1982). *The limits of obligation*. New Haven, CT: Yale University Press.

Flanagan, M., Howe, D. C., & Nissenbaum, H. (2005). Values at play: Design tradeoffs in socially-oriented game design. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 751–760), New York, NY.

Floyd, C., Mehl, W., Reisin, F., Schmidt, G., & Wolf, G. (1989). Out of Scandinavia: Alternative approaches to software design and system development. *Human-Computer Interaction, 4*(4), 253–349.

Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., et al. (2005). Predicting human interruptability with sensors. *ACM Transactions on Computer-Human Interaction, 12*(1), 119–146.

Fogg, B. J., & Tseng, H. (1999). The elements of computer credibility. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 1999*) (pp. 80–87), New York, NY.

Foot, P. (1978). *Virtues and vices*. Berkeley, CA: University of California Press.

Frankel, M. S., & Siang, S. (1999, November). *Ethical and legal aspects of human subjects research on the internet*. Washington, DC: American Association for the Advancement of Science. Workshop report. Retrieved March 15, 2007, from http://www.aaas.org/spp/sfrl/projects/intres/report.pdf

Friedman, B. (Ed.). (1997a). *Human values and the design of computer technology*. New York: Cambridge University Press.

Friedman, B. (1997b). Social judgments and technological innovation: Adolescents' understanding of property, privacy, and electronic information. *Computers in Human Behavior, 13*(3), 327–351.

Friedman, B. (1999). *Value-sensitive design: A research agenda for information technology* (Contract No.: SBR-9729633). Report to the National Science Foundation. Retrieved March 15, 2007, from http://www.ischool.washington.edu/vsd/files/friedman99VSD_Research_Agenda.pdf

Friedman, B. (2004). Value sensitive design. In *Encyclopedia of human-computer interaction* (pp. 769–774). Great Barrington, MA: Berkshire Publishing Group.

Friedman, B., & Borning, A. (2001). Value-sensitive design: Cultivating research and community (Contract No.: IIS-0000567). Unpublished report to the National Science Foundation.

Friedman, B., Freier, N. G., & Kahn, P. H., Jr. (2004). Office window of the future?—Two case studies of an augmented window. *Extended Abstracts of the Conference on Human Factors in Computing Systems,* New York, NY, 1559.

Friedman, B., Howe, D. C., & Felten, E. (2002). Informed consent in the Mozilla browser: Implementing value-sensitive design [Abstract]. *Proceedings of the Thirty-Fifth Annual Hawai'i International Conference on Systems Science,* Los Alamitos, CA, 247.

Friedman, B., & Kahn, P. H., Jr. (1992). Human agency and responsible computing: Implications for computer system design. *Journal of Systems Software, 17*, 7–14.

Friedman, B., Kahn, P. H., Jr., & Borning, A. (in press). Value sensitive design and information systems. In P. Zhang & D. Galletta (Eds.), *Human-computer interaction in management information systems: Foundations*. Armonk, NY: M. E. Sharpe.

Friedman, B., Kahn, P. H., Jr., Hagman, J., Severson, R. L., & Gill, B. (in press). The watcher and the watched: Social judgments about privacy in a public place. *Human-Computer Interaction.*

Friedman, B., Kahn, P. H., Jr., & Howe, D. C. (2000). Trust online. *Communications of the ACM, 43*(12), 34–40.

Friedman, B., Lin, P., & Miller, J. K. (2005). Informed consent by design. In L. Cranor and S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 495–521). Cambridge, MA: O'Reilly and Associates.

Friedman, B., & Millett, L. (1995). "It's the computer's fault"—Reasoning about computers as moral agents. *Conference Companion of the Conference on Human Factors in Computing Systems* (*CHI 95*) (pp. 226–227), New York, NY.

Friedman, B., & Nissenbaum, H. (1996). Bias in computer systems. *ACM Transactions on Information Systems, 14*(3), 330–347.

Friedman, B., & Nissenbaum, H. (1997). Software agents and user autonomy. *Proceedings of the First International Conference on Autonomous Agents* (pp. 466–469), New York, NY.

Fuchs, L. (1999). AREA: A cross-application notification service for groupware. In *Proceedings of the 6th European Conference on Computer Supported Cooperative Work* (*ECSCW 1999*) (pp. 61–80). Dordrecht, The Netherlands.

Galegher, J., Kraut, R. E., & Egido, C. (Eds.). (1990). *Intellectual teamwork: Social and technological foundations of cooperative work*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gewirth, A. (1978). *Reason and morality*. Chicago, IL: University of Chicago Press.

Godefroid, P., Herbsleb, J. D., Jagadeesan, L., & Li, D. (2000). Ensuring privacy in presence awareness systems: Automated verification approach. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*) (pp. 59–68), New York, NY.

Goldberg, K. (Ed.). (2000). *The robot in the garden: Telerobotics and telepistemology on the Internet*. Cambridge, MA: The MIT Press.

Good, N. S., & Krekelberg, A. (2003). Usability and privacy: A study of Kazaa P2P file-sharing. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 137–144), New York, NY.

Gotterbarn, D. (1999, November/December). How the new software engineering code of ethics affects you. *IEEE Software, 16*(6), 58–64.

Greenbaum, J. (1996). Back to labor: Returning to labor process discussions in the study of work. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 1996*) (pp. 315–324), New York, NY.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Greenspan, S., Goldberg, D., Weimer, D., & Basso, A. (2000). Interpersonal trust and common ground in electronically mediated communication. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*) (pp. 251–260), New York, NY.

Grief, I. (Ed.). (1988). *Computer-supported cooperative work: A book of readings*. San Mateo, CA: Morgan Kaufmann.

Gross, M. D., Parker, L., & Elliot, A. M. (1997). MUD: Exploring tradeoffs in urban design. *Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures* (pp. 373–387), Dordrecht, The Netherlands.

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW–88*) (pp. 85–93), New York, NY.

Habermas, J. (1979). *Communication and the evolution of society.* Boston, MA: Beacon Press. (Original work published, 1976.)

Hagman, J., Hendrickson, A., & Whitty, A. (2003). What's in a barcode? Informed consent and machine scannable driver licenses. *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 912–913), New York, NY.

Harris Poll/*Business Week*. (2000). *A growing threat.* Retrieved March 15, 2007, from http://www.businessweek.com/2000/00_12/b3673010.htm

Harrison, S., & Dourish, P. (1996). Re-place-ing space: The roles of place and space in collaborative systems. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 96*) (pp. 67–76), New York, NY.

Hatch, E. (1983). *Culture and morality*. New York: Columbia University Press.

Helwig, C. C. (2006). Rights, civil liberties, and democracy across cultures. In M. Killen & J. G. Smetana (Eds.), *Handbook of moral development* (pp. 185–210). Mahwah, NJ: Lawrence Erlbaum.

Herskovits, M. J. (1952). *Economic anthropology: A study of comparative economics*. New York: Alfred A. Knopf.

Hert, C. A., Liddy, E. D., Shneiderman, B., & Marchionini, G. (2003). Supporting statistical electronic table usage by citizens. *Communications of the ACM, 46*(1).

Höök, K. (1998). Evaluting the utility and usability of an adaptive hypermedia system. *Knowledge-Based Systems, 10*, 311–319.

Houston, J. (1995). *Confessions of an igloo dweller*. New York: Houghton Mifflin.

Hudson, J. M., & Bruckman, A. (2005). Using empirical data to reason about Internet research ethics. *Proceedings of the 9th European Conference on Computer-Supported Cooperative Work* (pp. 287–306), Dordrecht, The Netherlands.

Hudson, J. M., Christensen, J., Kellog, W. A., & Erickson, T. (2002). "I'd be overwhelmed, but it's just one more thing to do": Availability and interruption in research management. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2002*), New York, NY, 97–104.

Hudson, S. E., & Smith, I. (1996). Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work* (*CSCW 96*) (pp. 248–257), New York, NY.

Hughes, T. P. (1994). Technological momentum. In M. R. Smith & L. Marx (Eds.), *Does technology drive history? The dilemma of technological determinism* (pp. 101–113). Cambridge, MA: The MIT Press.

Iachello, G., & Abowd, G. D. (2005). Privacy and proportionality: Adapting legal evaluation techniques to inform design in ubiquitous computing. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2005*) (pp. 91–100), New York, NY.

Iachello, G., Smith, I., Consolvo, S., Abowd, G. D., Hughes, J., Howard, J., et al. (2005). Control, deception, and communication: Evaluating the deployment of a location-enhanced message service. *Proceedings of UbiComp 2005* (pp. 213–231), New York, NY.

Iacono, S., & Kling, R. (1987). Changing office technologies and the transformation of clerical jobs. In R. Kraut (Ed.), *Technology and the transformation of white collar work*. Hillsdale, NJ: Lawrence Erlbaum.

Isaacs, E. A., Tang, J. C., & Morris, T. (1996). Piazza: A desktop environment supporting impromptu and planned interactions. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 96*) (pp. 315–324), New York, NY.

Jacko, J. A., Dixon, M. A., Rosa, R. H., Jr., Scott, I. U., & Pappas, C. J. (1999). Visual profiles: A critical component of universal access. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 99*) (pp. 330–337), New York, NY.

Jancke, G., Venolia, G. D., Grudin, J., Cadiz, J. J., & Gupta, A. (2001). Linking public spaces: Technical and social issues. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 2001*) (pp. 530–537), New York, NY.

Jiang, X., Hong, J. I., & Landay, J. A. (2002). Approximate information flows: Socially-based modeling of privacy in ubiquitous computing. *Proceedings of UbiComp 2002* (pp. 176–193), New York, NY.

Johnson, D. G. (1985). *Computer ethics* (1st ed.). Englewood Cliffs, NJ: Prentice Hall.

Johnson, D. G. (2001). *Computer ethics* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Johnson, D. G., & Miller, K. (1997). Ethical issues for computer scientists and engineers. In A. B. Tucker, Jr. (Ed.), *The computer science and engineering handbook* (pp. 16–26). Boca Raton, FL: CRC Press.

Jones, S., Wilkens, M., Morris, P., & Masera, M. (2000). Trust requirements in e-business. *Communications of the ACM, 43*(12), 80–87.

Kahn, P. H., Jr. (1991). Bounding the controversies: Foundational issues in the study of moral development. *Human Development, 34*, 325–340.

Kahn, P. H., Jr. (1992). Children's obligatory and discretionary moral judgments. *Child Development, 63*, 416–430.

Kahn, P. H., Jr. (1999). *The human relationship with nature: Development and culture*. Cambridge, MA: MIT Press.

Kahn, P. H., Jr. (2006). Nature and moral development. In M. Killen & J. G. Smetana (Eds.), *Handbook of moral development* (pp. 461–480). Mahwah, NJ: Lawrence Erlbaum Associates.

Kahn, P. H., Jr., Freier, N. G., Friedman, B., Severson, R. L., & Feldman, E. (2004). Social and moral relationships with robotic others? *Proceedings of the 13th International Workshop on Robot and Human Interactive Communication* (*RO-MAN '04*) (pp. 545–550), Piscataway, New Jersey.

Kahn, P. H., Jr., & Friedman, B. (in press). *Augmented reality of the natural world and its psychological effects: A value-sensitive design approach*.

Kahn, P. H., Jr., & Kellert, S. R. (Eds.). (2002). *Children and nature: Psychological, sociocultural, and evolutionary investigations*. Cambridge, MA: MIT Press.

Kahn, P. H., Jr., & Lourenço, O. (1999). Reinstating modernity in social science research—or—The status of Bullwinkle in a post-postmodern era. *Human Development, 42*, 92–108.

Kahn, P. H., Jr., & Turiel, E. (1988). Children's conceptions of trust in the context of social expectations. *Merrill-Palmer Quarterly, 34*, 403–419.

Kant, I. (1964). *Groundwork of the metaphysic of morals* (H. J. Paton, Trans.). New York: Harper Torchbooks. (Original work published 1785.)

Kensing, F., & Madsen, K. H. (1991). Generating visions: Future workshops and metaphorical design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 155–168). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kensing, F., Simonsen, J., & Bødker, K. (1998). MUST: A method for participatory design. *Human-Computer Interaction, 13*(2), 167–198.

Kensing, F, Simonsen, J., & Bødker, K. (2004). Participatory IT design—An exemplary case. *Journal of the Center for Information Studies, 5*(3), 58–68.

Kiesler, S. (Ed.). (1997). *The culture of the Internet*. Mahwah, NJ: Lawrence Erlbaum Associates.

King, J. L. (1983). Centralized versus decentralized computing: Organizational considerations and management options. *Computing Surveys, 15*(4), 320–349.

Kling, R. (1980). Social analyses of computing: Theoretical perspectives in recent empirical research. *Computing Surveys, 12*(1), 61–110.

Kling, R. (1999). What is social informatics and why does it matter? *D-Lib Magazine, 5*(1). Retrieved March 15, 2007, from http://www.dlib.org/dlib/january99/kling/oikling.html

Kling, R., Rosenbaum, H., & Hert, C. (1998). Social informatics in information science: An introduction. *Journal of the American Society for Information Science, 49*(12), 1047–1052.

Kling, R., & Star, S. L. (1998). Human centered systems in the perspective of organizational and social informatics. *Computers and Society, 28*(1), 22–29.

Korpela, M., Soriyan, H. A., Olufokunbi, K. C., Onayade, A. A., Davies-Adetugbo, A., & Adesanmi, D. (1998). Community participation in health informatics in Africa: An experiment in tripartite partnership in Ile-Ife, Nigeria. *CSCW Journal, 7*(3–4), 339–358.

Kuhn, S., & Winograd, T. (1996). Profile 14: Participatory design. In T. Winograd (Ed.), *Bringing design to software* (pp. 273–294). Reading, MA: Addison-Wesley.

Kyng, M., & Mathiassen, L. (Eds.). (1997). *Computers and design in context*. Cambridge, MA: The MIT Press.

Lazar, J., Dudley-Sponaugle, A., & Greenidge, K. (2004). Improving web accessibility: A study of webmaster perceptions. *Computers and Human Behavior, 20*(2), 269–288.

Latour, B. (1992). Where are the missing masses? The sociology of a few mundane artifacts. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 225–258). Cambridge, MA: The MIT Press.

Leveson, N. G. (1991). Software safety in embedded computer systems. *Communications of the ACM, 34*(2), 34–46.

Leveson, N. G., & Turner, C. S. (1993). An investigation of the Therac-25 accidents. *IEEE Computer, 26*(7), 18–41.

Lipinski, T. A., & Britz, J. J. (2000). Rethinking the ownership of information in the 21st century: Ethical implications. *Ethics and Information Technology, 2*(1), 49–71.

Louden, R. B. (1984). On some vices of virtue ethics. *American Philosophical Quarterly, 21*, 227–235.

MacIntyre, A. (1984). *After virtue*. Notre Dame, IN: University of Notre Dame Press.

Mackay, W. E. (1995). Ethics, lies and videotape. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 95*) (pp. 138–145), New York, NY.

Mackay, W. E., & Fayard, A. L. (1999). Designing interactive paper: Lessons from three augmented reality projects. *Proceedings of IWAR '98* (pp. 81–90), Natick, MA.

Mackie, J. L. (1977). *Ethics: Inventing right and wrong*. New York: Penguin Books.

Malone, T. W. (1994). Commentary on Suchman article and Winograd response. *CSCW Journal, 3*, 37–38.

Mankoff, J., Fait, H., & Tran, T. (2005). Is your web page accessible?: A comparative study of methods for assessing web page accessibility for the blind. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2005*) (pp. 41–50), New York, NY.

McCarthy, J. F., & Anagnost, T. D. (1998). MusicFX: An arbiter of group preferences for computer supported collaborative workouts. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 98*) (pp. 363–372), New York, NY.

Milewski, A. E., & Smith, T. M. (2000). Providing presence cues to telephone users. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*) (pp. 89–96), New York, NY.

Millett, L., Friedman, B., & Felten, E. (2001). Cookies and web browser design: Toward realizing informed consent online. *Proceedings of the Conference on Human Factors in Computer Systems* (*CHI 2001*) (pp. 46–52), New York, NY.

Molich, R. (2001). Ethics in HCI. In *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2001*) (pp. 231–232), New York, NY.

Moor, J. H. (1985). What is computer ethics? *Metaphilosophy, 16*(4), 266–275.

Muller, M. J. (1997). Ethnocritical heuristics for reflecting on work with users and other interested parties. In M. Kyng & L. Mathiassen (Eds.), *Computers and design in context* (pp. 349–380). Cambridge, MA: The MIT Press.

Mynatt, E. D., Adler, A., Ito, M., Linde, C., & O'Day, V. L. (1999). The network communities of SeniorNet. *Proceedings of the Sixth European Conference on Computer Supported Cooperative Work* (pp. 219–238), Dordrecht, The Netherlands.

Nagel, T. (1986). *The view from nowhere*. Oxford, UK: Oxford University Press.

Nass, C., & Brave, S. (2005). *Wired for speech: How voice activates and advances the human-computer relationship.* Cambridge, MA: The MIT Press.

The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. (1978). *The Belmont report: Ethical principles and guidelines for the protection of human subjects of research*. Washington, DC: Author.

Neumann, P. G. (1991). Inside risks: Certifying professionals. *Communications of the ACM, 34*(2), 130.

Neumann, P. G. (1995). *Computer related risks*. New York: ACM Press.

Neustaedter, C., & Greenberg, S. (2003). The design of a context-aware home media space for balancing privacy and awareness. *Proceedings of UbiComp 2003*, (pp. 297–314). New York, NY.

Nielsen, J. (1993). *Usability engineering*. Boston, MA: AP Professional.

Nissenbaum, H. (1999). Can trust be secured online? A theoretical perspective. *Etica e Politica, 2*. Retrieved March 15, 2007, from http://www.univ.trieste.it/~etica/1999_2/nissenbaum.html

Nissenbaum, H. (2004). Privacy as contextual integrity. *Washington Law Review, 79*(1), 119–158.

Noble, D. D. (1991). *The classroom arsenal: Military research, information technology and public education*. London: Falmer Press.

Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic Books.

Olson, J. S., & Olson, G. M. (2000). i2i trust in e-commerce. *Communications of the ACM, 43*(12), 41–44.

Olson, J. S., & Teasley, S. (1996). Groupware in the wild: Lessons learned from a year of virtual collaboration. *Proceedings of the Conference*

on Computer Supported Cooperative Work (*CSCW 96*) (pp. 419–427), New York, NY.

O'Neill, D. K., & Gomez, L. M. (1998). Sustaining mentoring relationships on-line. In *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 98*) (pp. 325–334), New York, NY.

Orlikowski, W. J. (1993). Learning from *Notes*: Organizational issues in groupware implementation. *The Information Society, 9*(3), 237–250.

Orlikowski, W. J. (2000). Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science, 11*(4), 404–428.

Oviatt, S. (1999). Mutual disambiguation of recognition errors in a multimodal architecture. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 99*) (pp. 576–583), New York, NY.

Palen, L., & Dourish, P. (2003). Unpacking "privacy" for a networked world. In *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 129–136), New York, NY.

Palen, L., & Grudin, J. (2002). Discretionary adoption of group support software: Lessons from calendar applications. In B. E. Munkvold (Ed.), *Implementing collaboration technologies in industry: Case examples and lessons learned* (pp. 159–180). London, UK: Springer-Verlag.

Palen, L., Salzman, M., & Youngs, E. (2000). Going wireless: Behavior & practice of new mobile phone users. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 2000*) (pp. 201–210), New York, NY.

Parker, D. (1979). *Ethical conflicts in computer science and technology*. Arlington, VA: AFIPS Press.

Parker, D., Swope, S., & Baker, B. (1990). *Ethical conflicts in information and computer science, technology, and business*. Wellesley, MA: QED Information Sciences.

Patil, S., & Lai, J. (2005). Who gets to know what when: Configuring privacy permissions in an awareness application. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 101–110), New York, NY.

Pelto, P. J. (1973). *The snowmobile revolution: Technology and social change in the arctic.* Menlo Park, CA: Cummings Publishing Company.

Perdue, P. C. (1994). Technological determinism in agrarian societies. In M. R. Smith & L. Marx (Eds.), *Does technology drive history? The dilemma of technological determinism* (pp. 169–200). Cambridge, MA: The MIT Press.

Perry, J., Macken, E., Scott, N., & McKinley, J. L. (1997). Disability, inability and cyberspace. In B. Friedman (Ed.), *Human values and the design of computer technology* (pp. 65–89). New York: Cambridge University Press.

Persson, P. Blom, J., & Jung, Y. (2005). DigiDress: A field trial of an expressive social proximity application. In *Proceedings of UbiComp 2005* (pp. 195–212). New York: Springer.

Phillips, D. J. (1998). Cryptography, secrets, and structuring of trust. In P. E. Agre & M. Rotenberg (Eds.), *Technology and privacy: The new landscape* (pp. 243–276). Cambridge, MA: The MIT Press.

Poltrock, S. E., & Grudin, J. (1994). Organizational obstacles to interface design and development. *ACM Transactions on Computer-Human Interaction, 1,* 52–80.

President's Information Technology Advisory Committee (PITAC). (1999, February 24). *Information technology research: Investing in our future* (Advance Copy). National Coordination Office for Computing, Information and Communications. Washington, DC. Retrieved March 15, 2007, from http://www.nitrd.gov/pitac/report/pitac_report.pdf

Putnam, R., Leonardi, R., & Nanetti, R. (1993). *Making democracy work: Civic traditions in modern Italy*. Princeton, NJ: Princeton University Press.

Rawls, J. (1971). *A theory of justice*. Cambridge, MA: Harvard University Press.

Reeves, B., & Nass, C. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. New York: Cambridge University Press.

Riegelsberger, J., Sasse, M. A., & McCarthy, J. D. (2003). Shiny happy people building trust?: Photos on e-commerce websites and consumer trust. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 121–128), New York, NY.

Rocco, E. (1998). Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 98*) (pp. 496–502), New York, NY.

Rorty, R. (1982). *Consequences of pragmatism*. Minneapolis, MN: University of Minnesota Press.

Rosenberg, S. (1997). Multiplicity of selves. In R. D. Ashmore & L. Jussim (Eds.), *Self and identity* (pp. 23–45). New York: Oxford University Press.

Rotenberg, M. (2000). *The privacy law sourcebook 2000: United States law, international law, and recent developments*. Washington, DC: Electronic Privacy Information Center.

Roth, A. E. (1990). New physicians: A natural experiment in market organization. *Science, 250*, 1524–1528.

Sawyer, S., & Rosenbaum, H. (2000). Social informatics in the information sciences: Current activities and emerging direction. *Informing Science, 3*(2), 89–95.

Scheffler, S. (1982). *The rejection of consequentialism*. Oxford, UK: Oxford University Press.

Schiano, D. J., & White, S. (1998). The first noble truth of cyberspace: People are people (even when they MOO). *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 98*) (pp. 352–359), New York, NY.

Schneider, F. B. (Ed.). (1999). *Trust in cyberspace*. Washington, DC: National Academy Press.

Schoeman, F. D. (Ed.). (1984). *Philosophical dimensions of privacy: An anthology*. Cambridge, UK: Cambridge University Press.

Scranton, P. (1994). Determinism and indeterminacy in the history of technology. In M. R. Smith & L. Marx (Eds.), *Does technology drive history? The dilemma of technological determinism* (pp. 143–168). Cambridge, MA: The MIT Press.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences, 3*, 417–458.

Searle, J. R. (1983, October 27). The word turned upside down. *New York Review of Books*, 7479.

Sengers, P., Boehner, K., David, S., & Kaye, J. (2005). Reflective design. *Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility* (pp. 49–58), New York, NY.

Sharp, L. (1980). Steel axes for stone-age Australians. In J. P. Spradley & D. W. McCurdy (Eds.), *Conformity and conflict* (pp. 345–359). Boston: Little, Brown, & Company. (Reprinted from *Human Organization*, *1952*, *11*, 17–22.)

Shneiderman, B. (2000). Universal usability. *Communications of the ACM, 43*(5), 84–91.

Shneiderman, B. (2003). Promoting universal usability with multi-layer interface design. In *Proceedings of the Conference on Universal Usability* (pp. 1–8). New York: ACM Press.

Shneiderman, B., & Rose, A. (1997). Social impact statements: Engaging public participation in information technology design. In B. Friedman (Ed.), *Human values and the design of computer systems* (pp. 117–133). New York: Cambridge University Press.

Shoemaker, G. B. D., & Inkpen, K. M. (2001). Single display privacyware: Augmenting public displays with private information. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 2001*) (pp. 522–529). New York, NY.

Shweder, R. A., Mahapatra, M., & Miller, J. B. (1987). Culture and moral development. In J. Kagan & S. Lamb (Eds.), *The emergence of moral-*

ity in young children (pp. 1–82). Chicago: University of Chicago Press.

Smith, M. R. (1994). Technological determinism in American culture. In M. R. Smith & L. Marx (Eds.), *Does technology drive history? The dilemma of technological determinism* (pp. 1–35). Cambridge, MA: The MIT Press.

Smith, M. R., & Marx, L. (Eds.). (1994). *Does technology drive history? The dilemma of technological determinism*. Cambridge, MA: The MIT Press.

Spinello, R. A., & Tavani, H. T. (2001). The Internet, ethical values, and conceptual frameworks: An introduction to cyberethics. *Computers and Society, 31*(2), 5–7.

Sproull, L., & Kiesler, S. (1991). *Connections: New ways of working in the networked organization.* Cambridge, MA: The MIT Press.

Stephanidis, C. (Ed.). (2001). *User interfaces for all: Concepts, methods, and tools*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sturgeon, N. L. (1988). Moral explanations. In G. Sayre-McCord (Ed.), *Essays on moral realism* (pp. 229–255). Ithaca, NY: Cornell University Press.

Suchman, L. (1994). Do categories have politics? The language/action perspective reconsidered. *CSCW Journal, 2*(3), 177–190.

Tan, D. S., & Czerwinski, M. (2003). Information voyeurism: Social impact of physically large displays on information privacy. *Extended Abstracts of the Conference on Human Factors in Computing Systems* (*CHI 2003*) (pp. 748–749). New York, NY.

Tang, J. C. (1997). Eliminating a hardware switch: Weighing economics and values in a design decision. In B. Friedman (Ed.), *Human values and the design of computer technology* (pp. 259–269). New York: Cambridge University Press.

Thomas, J. C. (1997). Steps toward universal access within a communications company. In B. Friedman (Ed.), *Human values and the design of computer systems* (pp. 271–287). New York: Cambridge University Press.

Turiel, E. (1983). *The development of social knowledge*. Cambridge, UK: Cambridge University Press.

Turiel, E. (1998). Moral development. In N. Eisenberg (Ed.), *Social, emotional, and personality development: Volume 3 of W. Damon* (*Ed.*), *Handbook of child psychology* (5th ed/pp. 863–932). New York, NY: Wiley.

Turiel, E. (2002). *The Culture of morality: Social development and social opposition*. Cambridge, UK: Cambridge University Press.

Turiel, E., Killen, M., & Helwig, C. C. (1987). Morality: Its structure, functions and vagaries. In J. Kagan & S. Lamb (Eds.), *The emergence of morality in young children* (pp. 155–244). Chicago: University of Chicago Press.

Turkle, S. (1996). *Life on the screen: Identify in the age of the Internet*. New York: Simon and Schuster.

Ulrich, R. S. (1984). View through a window may influence recovery from surgery. *Science, 224*, 420–421.

Ulrich, R. S. (1993). Biophilia, biophobia, and natural landscapes. In S. R. Kellert & E. O. Wilson (Eds.), *The biophilia hypothesis* (pp. 73–137). Washington, DC: Island Press.

Van House, N. A., Butler, M. H., & Schiff, L. R. (1998). Cooperative knowledge work and practices of trust: Sharing environmental planning data sets. *Proceedings of the Conference on Computer Supported Cooperative Work* (*CSCW 98*) (pp. 335–343), New York, NY.

Voida, A., Newstetter, W. C., & Mynatt, E. D. (2002). When conventions collide: The tensions of instant messaging attributed. *Proceedings of the Conference on Human Factors in Computing Systems* (*CHI 2002*) (pp. 187–194), New York, NY.

Waddell, P., Borning, A., Noth, M., Freier, N., Becke, M., & Ulfarsson, G. (2003). Microsimulation of urban development and location choices: Design and implementation of UrbanSim. *Networks and Spatial Economics, 3*(1), 43–67.

Wainryb, C. (1995). Reasoning about social conflicts in different cultures: Druze and Jewish children in Israel. *Child Development, 66*, 390–401.

Wainryb, C. (2006). Moral development in culture: diversity, tolerance, and justice. In M. Killen & J. G. Smetana (Eds.), *Handbook of moral development* (pp. 211–240). Mahwah, NJ: Lawrence Erlbaum.

Weckert, J., & Adeney, D. (1997). *Computer and information ethics*. Westport, CT: Greenwood Press.

Weiser, M., & Brown, J. S. (1997). The coming age of calm technology. In P. Denning & B. Metcalfe (Eds.), *Beyond calculation: The next 50 years of computing* (pp. 75–85). New York: Springer-Verlag.

Weizenbaum, J. (1972, May 12). On the impact of the computer on society: How does one insult a machine? *Science, 178*, 609–614.

Wiener, N. (1985). The machine as threat and promise. In P. Masani (Ed.), *Norbert Wiener: Collected works and commentaries* (Vol. IV, pp. 673–678). Cambridge, MA: The MIT Press. (Reprinted from *St. Louis Post Dispatch*, December 13, 1953).

Williams, B. (1985). *Ethics and the limits of philosophy*. Cambridge, MA: Harvard University Press.

Winner, L. (1986). *The whale and the reactor: A search for limits in an age of high technology*. Chicago: University of Chicago Press.

Winograd, T. (1994). Categories, disciplines, and social coordination. *CSCW Journal, 2*(3), 191–197.

Zheng, J., Veinott, E., Bos, N. D., Olson, J. S., & Olson, G. M. (2002). Trust without touch: Jumpstarting long-distance trust with initial social activities. *Proceedings of the Conference of Human Factors in Computing Systems* (*CHI 02*) (pp. 141–146), New York, NY.

·64·

COST JUSTIFICATION

J. Patrick Williams and Randolph G. Bias
University of Texas, Austin

Deborah J. Mayhew
Deborah J. Mayhew and Associates

Introduction **1268**
Justification, of Cost Justification **1268**
The ROI Is What It Is 1268
The Empirical Approach to User-Centered Design **1268**
To Whom Do Benefits Accrue? **1269**
Quantifying the Costs (Easy!) **1269**
Quantifying the Benefits (Harder) **1269**
The Value of Money—Various Financial Methods **1269**
Raw Dollars 1269
Payback Period 1270
Internal Rate of Return (IRR) 1270
Net Present Value 1270
Not "If," But "Which" **1270**
Why a Cost-Justification Approach Is More Important for Web Interfaces **1271**
A Spreadsheet Model for Custom-Building Your Own Cost-justification Tool **1272**
Savings to the Customer **1275**
Increased Conversion Rates **1275**
Some Final Points **1276**
What of the Not-For-Sale, Internal-Use Software or Website? 1276
Are We Worried About Achieving a Particular ROI, or About Avoiding a Disaster (and a Lawsuit)? 1276
The "Developers are Users" Inhibitor—How Shall We Learn Not to Depend on Our Own Intuitions? . . . 1276
Finding the Stakeholder Where the Cost and Benefit Lines Cross—What to Do When the Costs Outweigh the Benefits for a Particular Suborganization 1276
Conclusion **1276**
References **1277**

INTRODUCTION

A friend recently shopping for a car was delighted to realize that much of her shopping, if not the final purchase, could be taken care of online, free of the long waits or pushy sales representatives she feared would find when visiting dealers' lots. However, once she began her shopping, she quickly realized that few, if any, of the manufacturer or dealer websites supported the tasks she envisioned carrying out. She was often forced to complete a multistep customization process involving choices among options packages, hubcap styles, and interior preferences before she was able to view even a basic price for a particular model. In some cases, she was required to enter personal information or set up an account by completing tedious forms in order to retrieve relevant pricing information. Unhappy with what she saw as artificial barriers to information she required to make her purchase (which, she admits, she would likely have conducted online), my friend began to resent the companies whose sites she felt wasted her time. *If this is how I'm treated in the privacy of my own home,* she wondered, *what should I expect if I were to walk on to the lot?* In frustration, my friend opted to contact her bank for help and ended up completing the majority of her shopping through the bank's website (and eventually chose the bank for financing). Her interactions with the actual dealer were brief and left her with the impression that they were happy to waste her time. In effect, the usability problems on the manufacturer and dealer websites strengthened her preexisting aversion to their services when the sites were likely intended and supposedly designed to do just the opposite. They also cost a dealership a financing customer and led to negative feedback from my friend in a number of online car-shopping forums.

Around the same time, an advertisement for a nationwide used-car retailer network was getting substantial airplay. In it, a gentleman, excited about purchasing his new car online, has just selected his car and is beginning the "checkout" process on a competing car dealer's Web site to have the automobile financed and delivered to his home. Then the look on his face tragically changes when he finds himself caught in the familiar infinite loop of attempting to guess an appropriate user ID and password for the site. As the frustrated potential customer continues to battle with the website, a reassuring voiceover promises viewers that their experience with the advertiser's website will be pleasant and easy. We are left wondering whether the gentleman was ever able to create a suitable username and carry on with his purchase.

As the stakes for online purchases rise higher and higher, a lost sale caused by poor usability can have an unprecedented negative business impact. Some companies and advertising firms have begun to leverage usability as a competitive advantage, and consumer willingness to accommodate frustration in a virtual sea of alternatives, as just illustrated, is low. How much revenue will the car dealers lose from situations like to ones you just read about? More interestingly, how many sales, online or otherwise, will poor usability cost these companies, and how many such losses are acceptable?

JUSTIFICATION, OF COST JUSTIFICATION

"The usability engineering of websites is *always* carried out under suboptimal conditions. Development constraints . . . demand that we pursue a quicker, less-thorough course of usability engineering than we would like" (Bias & Keough, 2000). And in times of tight budgets, it is absolutely necessary to be able to demonstrate, with real, quantitative data, that any money invested in usability engineering yield a positive return. Otherwise, the usability engineer is dependent upon the faith and inspiration of the stake-holders, when trying to get the funding to carry out a user-centered design approach. Plus, said usability engineer will be competing for those funds with others who DO have quantitative figures—lines of code, sales projections, market research, defect rates, customer support calls, person months.

Consider, too, that much of the software-development world still sees usability as something "nice to have." To compete for those constrained development resources, the usability engineer must be able to justify the cost of any expenditures on usability. This chapter is intended to give to the usability engineer, the usability engineering manager, or anyone advocating for the implementation of a user-centered design approach, the tools to demonstrate that a particular program of usability engineering will yield a particular return on investment (ROI).

The ROI Is What It Is

The ROI for any effort is what it is—the trick is to project or calculate it. Theoretically, after each new marketing campaign or addition in sales force or added product feature, we could calculate effects on revenues and costs. (This is not necessarily easy, as usually there are multiple variables with no experimental control, and thus it is hard to attribute a certain percentage of any change to one variable.) Given the state of web design, and our experiences, usability engineers are confident that just about all usability engineering efforts are cost justified. Clare-Marie Karat (2005) reported a range of ROI ratios from 2:1 to 100:1 for traditional software user interface designs. And with web-development tools enabling more folks to play the role of designer and the broad access to the web making for an ever-widening circle of users, there's every reason to suspect that usability engineering for websites yields an even HIGHER positive return.

So, assuming that any change in a company's product or process will yield SOME change in that company's revenues or expenses or both, the trick is to estimate what that ROI will be.

THE EMPIRICAL APPROACH TO USER-CENTERED DESIGN

Though it is hard to attribute exact fluctuations in costs and benefits to certain methods employed in a user-centered design approach, it is possible to collect empirical data to drive projections and calculations of ROI. In this chapter, we will en-

deavor to encourage and equip the usability engineer, or the software development manager, to collect the data and make the calculations to justify expenditures on usability engineering. We will begin by considering the wide range of possible beneficiaries of a user-centered design approach.

TO WHOM DO BENEFITS ACCRUE?

It is important to realize, and to communicate to all the product/Web site stakeholders, the different benefits that can result from a program of usability engineering. Done right (and early in the development cycle), usability engineering can yield efficiency gains for the *site/product-development team*, as problems are discovered early when they are relatively inexpensive to repair. The *customer- support team* will require a relatively small staff, as more users can carry out their tasks without help. Similarly, the *documentation team* may need to generate fewer user aids, and will do a better job of creating their help screens and other hard-copy and online documentation. A more intuitive product means that fewer resources need to be devoted to *training* materials—or perhaps the training products can focus more on advanced usage, as the fundamentals are more intuitive. Of course, a more usable product will mean that *the user* will enjoy increased throughput, both because the interaction design allows quicker completion of tasks through fewer displays and fewer keystrokes/mouse clicks, and because less time is spent making and recovering from errors. Indeed, *the customer* may be able to hire a less sophisticated or less experienced employ to carry out tasks previously handled only by higher-paid employees. All of this leads to increased customer satisfaction, increased trade-press joy, and increased sales.

QUANTIFYING THE COSTS (EASY!)

It is a simple matter to quantify the COSTS. For in-house usability teams, it is a simple matter to calculate the costs of the usability engineers—their salaries and additional costs such as benefits, a desk, a computer, and so forth. ("Commonly, the fully loaded cost of an employee is at least twice his or her salary" [Nielsen, 2000, p. 276].) For a particular project, or even a particular data-collection effort, a fractional cost of a usability professional can be calculated. For usability engineering contractors or consultants, the costs will be easy to calculate, simply by reading the invoice. You must also add to these costs the costs of engaging the software developers in the usability evaluations or in the implementation of the corrections to the interface that arose during testing. That is, perhaps the designer/developers will be asked to generate a user interface prototype for testing. Or perhaps they will observe end-user testing in the lab. Certainly the development team will need to invest some time in helping build a user profile and test scenarios.

Recognize that there will be *some* costs, even without usability engineering. That is, when calculating the comparison between the costs of pursuing a usability engineering approach and not, it is important to include the designer/developer time to fix software or design bugs that may be discovered late in the process anyway, say, during system test.

QUANTIFYING THE BENEFITS (HARDER)

Quantifying the benefits is harder, but still possible. One reasonable approach is to make projections based on usability testing. Perhaps the crispest arena is reduced call-support burden. Imagine a usability test that revealed that of the test participants who carried out a task, 30% had to call the (simulated) help desk. And then (we're still imagining), in light of the usability test results, the product is redesigned, and subsequent testing reveals that only 5% of the test participants need help to complete the task. It is a relatively simple matter to project the reduction in the number of calls to the help desk using the projected number of licenses (for a product) or visits (for a website). Then multiply that number of *avoided* help-line calls times the average cost of a call to the company, to find the amount of money saved thanks to that one usability test. (We'll go through a more thorough example in a later section.)

Another way to project the benefits is by using historical data. If you have experience with a similar sort of interface at a similar stage of development, you might reasonably expect the same ROI as you earned the last time. Or if there are data in the research literature, from a project that is similar to yours, you could use those ROI data as an estimate. Of course, the closer your current situation matches that of the other development effort from which you're extrapolating—in complexity of the user interface, sophistication, and experience level of the development team, processes followed by the development team, resources available to the development team (e.g., machines for testing), domain expertise in house—the better the estimate will be. As with sales projections, it is always best to have some customer data to drive the estimates.

It is important to return to the "actuals" after the project to assess how good your ROI estimates were. This is a difficult thing to budget time for, as you will be off on the next project. However, the continuous improvement of your financial modeling and your empirically based validation of your usability effort, in the eyes of your stakeholders and funders, depends on your ability to demonstrate that your projections were good (and conservative!), and that the hour and dollar spent on usability engineering were well worth it.

THE VALUE OF MONEY—VARIOUS FINANCIAL METHODS

Raw Dollars

One of the frustrations with usability engineering of human-computer interfaces (HCI), circa 2005, is there is no acknowl-

edged metric with which all interfaces can be compared. Recent attempts have been made to boil down usability evaluation data to a single summative score (Sauro & Kindlund, 2005), but until such a method comes into widespread use, most usability engineers will be more comfortable relying on the universally understood metric historically at their disposal: the dollar. (Or the franc, or the yen; for a discussion of the application of a cost-justification approach to projects that span global markets, see Mayhew & Bias [2005]) It enables managers, engineers, writers, marketers, and even psychologists to communicate on even ground: if we spend X dollars on usability engineering, we expect to save Y dollars thanks to gained development efficiencies and reduced support call burden, and to earn Z additional dollars in revenue, thanks to increased sales. And as stated earlier, the benefits of usability engineering tend to be so robust that attending only to raw dollars will usually be sufficient to justify dedicating resources to a systematic, user-centered design approach.

Three different but related factors, however, may today motivate individuals and, in the near future, the whole discipline to employ more sophisticated financial models:

1. Tight development budgets and the need for usability engineers to compete with other product-related disciplines
2. The need to be smart about *which* usability-engineering methods to employ (more in the next section)
3. The (assumed) ever-improving accuracy of our projections of the benefits of usability engineering.

These three factors may drive the usability engineer intent on being as accurate as possible in projecting the relative value of usability to employ an accounting technique such as one of the following three. (See Karat [2005] for more detail, including formulae.)

Payback Period

The *payback period* is the amount of time it takes to recover through incremental cash flows (or subsequent cost reductions) a project's net investment cash outflow. The calculation locates the point at which the cash outflow equals the cash inflow. In simple terms, the shorter the payback period, the better the investment. An organization may have a standard payback period that any investment must beat in order to be funded. Alternately, two different courses of usability engineering (say, a "basic" course versus a "complete" course) can be compared against each other to drive a decision. Payback period is not a measure of profitability, but rather a measure of speed of capital recovery. (Note: It is *not* necessarily the case that the more expensive, "full" course will yield a longer payback period.) (For a passionate and well-reasoned argument for the joys of a "time-to-break-even" approach, see Conklin, [1996]).

Because payback period ignores the time value of money, many organizations use discounted cash-flow techniques to make investment decisions. Two of the most popular are (a) internal rate of return (IRR) and (b) net present value (NPV).

Internal Rate of Return (IRR)

The internal rate of return (IRR) on a project is the company's rate of return on invested capital. For example, if a particular project costs $1,000 and earns $100 in year 1 and 2 plus returning the initial investment at the end, than the IRR is 10%.

IRR

	Time O	**Year 1**	**Year 2**
Cash Flow	**($1,000)**	**$100**	**$1,100**
	10%	IRR	

The IRR in this case is compared against a company's cost of capital to make an investment decision. *Cost of capital,* in simple terms, is the company's cost of obtaining funding whether through the debt or equity markets. Most companies use both sources. For example, if a company's capital structure is 50% debt with an average cost of 10%, and 50% through equity with an average return of 20% the cost of capital is 15% (.5 ∗ .1) + (.5 ∗ .2). In summary, the decision rule for IRR is as follows: if the IRR is greater than the cost of capital, then the project should be accepted, but if the opposite is true, the project should be rejected. Net present value is an alternative to IRR. With IRR, we calculate a percentage, whereas with NPV we calculate a dollar amount.

Net Present Value

NPV is today's value of the cash inflows less the cash outflows. In this technique the future cash inflows are discounted using the company's cost of capital. From a finance view, any project with a positive NPV should be accepted. In the above example, with a cost of capital of 10%, the NPV is zero and we would be indifferent regarding the project.

NPV

	Time O	**Year 1**	**Year 2**
Cash Flow	**($1,000)**	**$100**	**$1,100**
	10%	Cost of Capital	
	($0.00)	NPV	

The challenge to any usability engineer is to demonstrate that the benefits in the form of cash inflows (today's dollars) outweigh the cost of the project as measured against the organization's cost of capital. In many companies, 15 to 20% is the hurdle rate used to determine whether or not a project is acceptable.

NOT "IF," BUT "WHICH"

As suggested in the last section, one can apply a cost-justification approach *not only* to justify (or not) usability engineering of a product interface, but (*more likely, these days*) to steer the product development team toward a *particular set* of usability engineering methods. The field of usability engineering has come to a point of sophistication where we needn't ask the question,

"Shall we employ usability engineering?," but rather we should ask, "Which methods will yield us the best ROI for our usability engineering dollar and hour?" It would be folly for a web or other software-development team to ask the question, "Shall we test the software before 'going live,' or before shipping." Obviously, the sensible question is, "How much software testing shall we employ and what type of testing"?

Likewise, given the importance of usability to the success of websites and just about all software, the sensible question is not, "Shall we employ usability engineering in the development of our product/site?" but rather, "How much usability engineering should we invest in, which methods should be employ, and when?" Clearly, $20,000 spent on a prototype test, early in the development cycle, may yield a greater ROI than a $20,000 end-user test after the product is developed, given the relative costs of repairing the problems discovered, even given the increased time-adjusted cost of the earlier test.

WHY A COST-JUSTIFICATION APPROACH IS MORE IMPORTANT FOR WEB INTERFACES

Another anecdote will serve to illustrate the increased importance of usability on websites as compared to traditional office software.

A contract development team was building a website for a client organization. The website was to include up-to-date drug information and was intended to be used by physicians as a substitute for the standard desk references they currently use to look up such drug information as side effects, interactions, appropriate uses, and data from clinical trials. The business model for the site was an advertising model. Physicians would visit the site regularly because more current information would be available and that information would be more easily found on the site and relative to published desk references. Pharmaceutical companies would buy advertising for their drug products on the site because the visitors to the site (physicians) represent their target market. Repeat visits would generate regular and new traffic and new visitors would join based on word-of-mouth among physicians. This would drive up the value of advertising generating a profit for the client.

The development team generated a prototype design that the client would use to pursue venture capital to support the full-blown development and initial launch and maintenance of the site. The client paid for this prototype development.

Once the prototype was ready, a usability engineering staff was brought on board to design and conduct a usability test. Eight physicians were paid to fly in to the development center city and participate in the test as test users. Several basic search tasks were designed for the physicians to perform. They were pointed to the prototype's Home Page and left on their own to try to successfully find the drug information that was requested in the first task.

Within 45 seconds of starting their first search task, seven out of the eight physicians gave up and announced that the site was unusable and if it were a real site, they would abandon it at that point and never return.

Clearly, if the site had launched as it was prior to this test, not only would an optimal ROI not have been realized, but also the site would have failed all together and the clients would have completely lost their investment. If only 7/8ths of all visitors did not return, there would not be enough traffic to motivate advertisers to buy advertising. The entire investment would have been lost.

Instead, the test users were asked to continue with the entire test protocol, and the data generated revealed insights into the problem that was a show-stopper on the first task, as well as other problems revealed in other test tasks. The site was redesigned to eliminate the identified problems. Clearly, the usability test, which had an associated cost, was worth the investment in this case.

This anecdote illustrates something that distinguishes websites from commercial software products. In a commercial software product, the buyers discover the usability problems only after they have paid for the product. Often they cannot return it once they have opened the shrink wrap and installed the product. Even if it has a money-back guarantee, they are not likely to return it, and even if they did return it, there may not be many alternative products on the market, which are easier to use.

On a website, on the other hand, it costs the visitor nothing to make an initial visit. On a website based on an advertising model, such as the one described, the site investor makes nothing at all unless there is sufficient ongoing traffic to attract and keep advertisers. On an e-commerce site, the investor makes nothing at all unless the visitors actually find and successfully purchase products.

A website is not a product and the user does not have to buy it to use it. The website is just a channel, like a TV show, magazine, or catalog, and if users do not find and repeatedly and successfully use the channel, the investor gets no ROI for having developed the channel. Thus, usability can absolutely make or break the ROI for a website even more so than for traditional software products.

It's also true that competition is even more dependent on relative usability on websites than is the case with traditional software products or sales channels. Someone wishing to buy a book may be inclined to buy from a particular bookstore that is easy to get to, even if it is not the best bookstore around. On the other hand, if a customer cannot easily find the desired book through the barnesandnoble.com website, they need not even get out of their chair to shop at a competitor's site instead. It's not enough to simply have a website that supports direct sales; your site must be more usable than the competition's site or you will lose business based on the relative usability of the selling channel alone. For example, 60% of a sample of consumers shopping for travel online stated that if they cannot find what they are looking for quickly and easily on one travel site, they will simply leave and try a competitor's site (Forrester Research, Inc., 2000b). As illustrated in this chapter's opening anecdotes, savvy advertisers have even begun to explicitly bundle Web site usability into their clients' campaigns in a manner that will increase both customer awareness and demand for usable interfaces, and thus make relative usability among competitors even more important.

In addition, if you are a catalog-order company such as LL Bean or Land's End, and your product is good but your website is bad, customers will not use your website and will resort to traditional sales channels (fax, phone) instead. This will result in a poor ROI for the website that was intended to be justified by relatively low costs compared with that of those more traditional channels.

Forrester reported a buy-to-look ratio of only about 2% on e-commerce websites (Forrester Research, Inc., 2000a), and while part of this may be due to other factors, usability is surely a large factor in this poor performance of websites as a sales channel.

Another Forrester Report (Forrester Research, Inc., 2001) cited other compelling statistics from a variety of studies. In one study, 65% of shopping attempts at a set of prominent e-commerce sites ended in failure. In another, 42% of users were unable to complete the job-application process on a set of corporate sites. Another study showed that journalists could find answers to only 60% of their questions on corporate public relations sites. And still another study showed that adding certain product information to e-commerce sites reduced product-related inquiries by more than 20%. Finally, Forrester pointed out that when Amazon.com let customer service slip in 2000, they fell behind their competitors in three categories for the first time. In this same report, Forrester noted that the companies they interviewed spent between $100,000 and $1 million on site redesigns, but few had any sense of which specific design changes paid off, if any. Clearly, usability engineering programs that are specifically aimed at such goals as increased success rate of shopping attempts, job-application completion, and information requests could have had a profound impact on the ROI of these redesign projects.

As illustrated by all these examples, usability engineering can have a profound impact on ROI in the case of websites, and even more dramatic cost-justification cases can be made for usability-engineering programs during web development as compared to during traditional software-development projects. A profit is made whenever a commercial software product is sold, regardless of how productive users are once they start to use that software product, but there simply is no ROI until and unless visitors to websites complete their tasks efficiently and successfully.

A SPREADSHEET MODEL FOR CUSTOM-BUILDING YOUR OWN COST-JUSTIFICATION TOOL

The following is an example of a cost-justification tool, built on a simple spreadsheet tool with the addition of some fairly simple macros. It is offered as a model for how the reader might generate a cost-justification tool to project his or her own.

The following section of the spreadsheet, in Table 64.1, represents the summary cost-benefit analysis. According to this summary, we expect to spend $212,974 in year one. These expenses include building a $33,000 usability lab, plus about $180,000 spent on usability-engineering support on a total of three products. The summary reveals that by the end of year two we expect a net gain, and by the end of year three we expect a total ROI of $519,048, for a net gain of $306,074. And, this is the result before any additional sales due to improved usability are factored in.

The details that fuel these summary figures are offered below.

TABLE 64.1. Usability Cost Justification—Software Manufacturer

	Year 1	Year 2	Year 3	Total
Cost				
Usability Lab	($33,000)			
Usability Lab Projects[1]	($179,974)			
Total Cost	**($212,974)**	**$0**	**$0**	**($212,974)**
Cost Savings				
Customer Support	$168,000	$168,000	$168,000	
Software Development	$15,048			
Additional Sales				
Total Cost Savings	**$183,048**	**$168,000**	**$168,000**	**$519,048**
Net	**($29,926)**	**$168,000**	**$168,000**	**$306,074**

Our example illustrates the cost benefit from employing usability in a software-development environment. First let's examine the costs. Tables 64.2 and 64.3 project the cost of a "basic" program of usability engineering and a "complete" program of usability engineering, respectively. The "complete" program has all the "basic" program has, plus a prototype usability test and the inclusion of a usability survey along with the product itself. In the interest of making the usability cost-benefit analysis as conservative as possible, an attempt is made to be as thorough as possible, accounting for all the meetings, the developer hours, and anything else that would not have been a cost had a program of usability engineering not been pursued.

The hourly cost for each development team member (Developer, Usability engineer, QA engineer, Customer Support professional) is intended to be a "fully loaded" salary, including benefits, computer support, etc.

Table 64.4 represents the costs of building a usability lab.

There are two main elements of costs in our example: (a) the lab build-out and (b) the conduct of a program of usability engineering. This example assumes that existing space is used for the lab; thus, the incremental cost consists of lab design, build out, and hardware. The cost of the conduct of a program of usability engineering represents the cost to run two basic and one expanded or "complete" usability project. That is, we are going to assume that in year one we are going to utilize the lab, and the usability engineers, to provide "basic" support to two product development efforts (say, websites) and "complete" support to one product development effort. Thus, adding the lab cost ($33,000) to the two basic support efforts (at $51,378 each, for a total of $102,756) and one complete support effort ($77,218) yields a total year one cost for usability engineering of $212,974, which we saw in Table 64.1. (Note, when building your own

TABLE 64.2. Cost of "Basic" Usability-Engineering Support

Activity	Cost Item	Units (hours)	Unit Cost	Total Cost
Generate wish list	Developer	3	70	$210
	Customer Support	3	56	168
User survey	Developer	16	70	1120
	Postage			50
Specification review	Usability engineer	8	70	560
Scope review	Usability engineer	4	70	280
Contents review	Usability engineer	8	70	560
Create usability plan	Usability engineer	8	70	560
Feature review	Usability engineer	8	70	560
Function-task analysis	Usability engineer	40	70	2800
	Developer	40	70	2800
	Travel			5000
Detailed feature review	Usability engineer	8	70	560
QA review	Usability engineer	4	70	280
	QA engineers	32	70	2240
Usability walkthrough	Usability engineer	32	70	2240
	Developer	104	70	7280
	Customers	48	30	1440
	Travel			6000
	Miscellaneous			100
Beta survey	Usability engineer	128	70	8960
	Developer	104	70	7280
	Postage			50
Product ship meeting	Usability engineer	4	70	280
Total				**$51,378**

TABLE 64.3. Cost of "Complete" Usability Engineering Support

Activity	Cost Item	Units (hours)	Unit Cost	Total Cost
Generate wish list	Developer	3	70	$210
	Customer Support	3	56	168
User survey	Developer	16	70	1120
	Postage			50
Generate UI prototype	Developer	160	70	11,200
Test UI prototype	Usability engineer	96	70	6720
	Test subjects	80	30	2400
	Developer	72	70	5040
	Miscellaneous			200
Specification review	Usability engineer	8	70	560
Scope review	Usability engineer	4	70	280
Contents review	Usability engineer	8	70	560
Create usability plan	Usability engineer	8	70	560
Feature review	Usability engineer	8	70	560
Function-task analysis	Usability engineer	40	70	2800
	Developer	40	70	2800
	Travel			5000
Detailed feature review	Usability engineer	8	70	560
QA review	Usability engineer	4	70	280
	QA engineers	32	70	2240
Usability walkthrough	Usability engineer	32	70	2240
	Developer	104	70	7280
	Customers	48	30	1440
	Travel			6000
	Miscellaneous			100
Beta survey	Usability engineer	128	70	8960
	Developer	104	70	7280
	Postage			50
Product ship meeting	Usability engineer	4	70	280
Include usability survey	Usability engineer	4	70	280
Total				**$77,218**

TABLE 64.4. Cost of Building a Usability Lab

Lab Feature	Units	Unit Cost	Total Cost
Lab design and equipment selection	160	35	5600
Carpenters and electricians	25	80	2000
One-way mirror	1	1900	1900
Video recording equipment	3	3500	10,500
Computers	4	2000	8000
Overhead projector	1	5000	5000
Total			**33,000**

cost-justification spreadsheet, you could add macros to dynamically change the costs as you increase the number of basic and complete programs you undertake.)

Now let's address the potential benefits (Table 64.5). In this example, we consider only the cost savings based on market costs for software development and customer-support personnel. The customer-support benefit is derived from reduced call support burden and the software development savings are a result of reduced rework or engineering hours. This example illustrates that usability would result in a positive net present value (NPV) of $153,455 or a payback period of two years, just considering the customer support and software development savings.

Notice that our spreadsheet example has a macro in it that allow the user (of the spreadsheet) to change the "cost of capital" (basically, how much it costs to spend the money, versus keeping it and earning interest on it) in increments of 1%, which automatically yields different calculations of the time value of the expenditures and savings.

Now let's look at that savings in more detail. We determine that due to problems discovered early (when they are cheaper to fix) rather than late, the software-development team will realize a savings of $15,048 in the first year (as seen in Table 64.1). That is, based on historical data, we expect that it will take software developers 215 fewer hours (at $70 per hour) to fix the problems.

As for the detail for the reduced call-support burden, our product, an e-commerce site, expects 20,000 visitors a year. (Note the conservative nature of the estimates. If our estimates are low, and our projections yield a positive ROI, then actual numbers that exceed the estimates will yield even higher actual ROI.) By working with the customer-support team and reviewing their (very detailed!) call logs, we calculate that 70% of the calls are for nondefect problems. That is, the code works "as spec'd," and users just can't figure out how to carry out their task. The majority of customer-support calls a software manufacturer will receive tend to be these how-to/nondefect issues. Table 64.6 reveals a further breakdown of these usage problems that usability engineering testing is likely to help resolve.

The customer-support team also knows (and knows well!), the cost of a call. In our example, that cost is $120. Given the number of visits, the cost of a call, and the percentage of "usage" problems, we can calculate that these calls cost the company $1,680,000 a year. If usability testing were able to reveal problems and be an agent of change to drive some fixes, enough to reduce the "usage" calls by 10%, the company would realize a savings of $168,000 a year.

As in Table 64.5, Table 64.6 has a macro that allows the usability engineer, based on ever-improving data, change the

TABLE 64.5. Financial Metrics

	Year 0	Year 1	Year 2	Year 3	Cost of Capital 20%
Cash Flow	($212,974)	$183,048	$168,000	$168,000	
Discounted Cash Flow	($212,974)	$152,540	$116,667	$97,222	
Net Present Value	$153,455				
Internal Rate of Return	64%				
Payback Period	2 Years				

TABLE 64.6. Savings Due to Reduced Call-Support Burden

Product	Total Cases	Documentation Change	Environment Problem	Documentation Questions	Installation Questions	Product Usage	Usability Sub-total
E-commerce Website	20,000	1%	9%	1%	10%	49%	70%

Usability Benefit **10%**

Product	Cost Per Call	Total Cost	Usability Support Questions	Cost of Usability Related Cases	Usability Engineering Savings	Revised Total Cost
E-commerce Website	$120	$2,400,000	70%	$1,680,000	$168,000	$2,232,000

"usability benefit" (i.e., the percentage of usage calls avoided) by 1% increments.

This completes the first example. As we saw in Table 64.1, we expect to spend $212,974 in the first year, and at the end of three years to have realized savings of $519,048, for a net savings of $306,074. In Table 64.5 we see that this investment has a payback period of two years, with an internal rate of return of 64%. If we learn, during the course of this project, that the cost of capital goes down, or that the usability benefit is actually higher than 10%, our projected benefit of the usability engineering work will go up.

SAVINGS TO THE CUSTOMER

The previous illustration was exceedingly conservative as it did not take into account all the benefits that might possibly derive (not even increased sales). In the following example illustrated in Table 64.7, we illustrate how to calculate potential benefits to the customer.

Let's say we have a customer company that is buying a software product from us that will be used by 500 employees. If this customer receives a more usable product, then the users of that product will require less training, spend less time making and recovering from errors, and will be quicker at traversing the correct path through the user interface.

Thanks to the program of usability engineering, productivity increases due to users being able to navigate the product menus/websites more efficiently. Conservatively (again), based on historical findings we expect to improve performance using this software tool by 30 seconds. A 30-second benefit for 500 users would result in $33,542 worth of savings in the course of a year. Error reduction is the savings associated with crashes or redoing any given task. A 15-second benefit, thanks to one less error to correct, would results in savings of $16,771 for the same 500 users. Naturally, a more user-friendly product is more intuitive and would result in less required training. At a training cost of $35/hour, a one-hour reduction would result in a $17,500 one-time savings. Therefore, one customer with 500 "seats" will enjoy a savings of over $50,000 a year thanks to our usability engineering effort.

INCREASED CONVERSION RATES

Here's another example. Critical to an e-commerce website's success is its conversion rate and ability to generate repeat business. Many of us have had unpleasant web experiences, whether it was difficulty in navigating the website, conflicts with the user's computing environment due to platform or browser issues, differences in connection speeds, or other access problems. In most of these situations the potential customers do not call or e-mail. Instead they revert to a traditional route or alternatively move to a competitor's website (and in the worst-case scenario, submit an angry review to an online shopping service like Epinions.com or Froogle.com, or publicly express their frustration in any number of other ways that will reach other potential customers). In the example of a physicians' reference website, the failure to employ usability would result in a 13% conversion rate (one out of eight users returning), or 87% of potential users deciding not to use the site.

What if we project that via usability testing we would yield a 50% increase in the conversion rate, raising it from 13% to 19%? The macro in the spreadsheet allows you to adjust the projection based on a different projected benefit. Further, the spreadsheet in Table 64.8 reveals that we expect to gain $40 per active user, in advertising dollars. A small (say, $10,000) usability test would yield a $25,000 increase in advertising dollars. In our

TABLE 64.7. Usability Cost Justification—Customer

	Year 1	Year 2	Year 3	Total
Cost Savings				
Productivity Improvement	$33,542	$33,542	$33,542	
User Error Reduction	$16,771	$16,771	$16,771	
Reduced Training	$17,500			
Total Cost Savings	**$67,813**	**$50,313**	**$50,313**	**$168,439**

TABLE 64.8. Conversion Rate Benefits

Users	Conversion Rate	Active Users	Advertising Dollars per User	Dollar Value of Subscribers	Lost Advertising Dollars
10,000	13%	1,250	$40	$50,000	$350,000
Usability Benefit	**50%**				



Users	Conversion Rate	Active Users	Advertising Dollars per User	Dollar Value of Subscribers	Lost Advertising Dollars
10,000	19%	1,875	$40	$75,000	$325,000

example above, the usability testing and subsequent site redesign increased the conversion rate by 700%! Clearly, the ROI projections based on that testing would be much, much higher, and the usability work well cost-justified.

SOME FINAL POINTS

What of the Not-For-Sale, Internal-Use Software or Website?

In this chapter our examples have focused on vended software and e-commerce sites. Similar cost-benefit analyses are possible for intranet sites or informational sites. "For intranet sites, efficiency, memorability, and error reduction become the most important usability attributes" (Nielsen, 2000, p. 274). Thus, employees' time spent locating documents on the company intranet site can be calculated, and the resultant improvements due to a well-designed, usable site demonstrated.

Are We Worried About Achieving a Particular ROI, or About Avoiding a Disaster (and a Lawsuit)?

We have worried about numbers—dollars in particular. But there are intangible costs and benefits to be considered, which may or may not translate into dollars. Could another cost of poor usability engineering be a bad review in the trade press? Or it could be a lawsuit, if someone thinks an actual or implied contract was not fulfilled because people couldn't use a (perhaps defect-free) user interface effectively? Or it could mean a picket line of angry users, picked up by the six o'clock news? Or customer hostility publicly expressed all over the Internet? When evaluating whether or not to pursue a course of usability engineering, it's important to consider not only what you'll get that you like, but also what you won't get that you wouldn't like.

The "Developers are Users" Inhibitor—How Shall We Learn Not to Depend on Our Own Intuitions?

Why has usability not been more warmly and universally embraced as a vital aspect of software user interface development? Why is it that "usability is one of the first things cut in a budget" (Rhodes, 2000, p. 4)? Perhaps it is because all developers are also users. Every software developer is also software user, and so reflexively feels justified to depend on his or her own intuitions as to what is usable. While the typical usability engineer wouldn't think of telling a software developer how to structure his or her code, the software developer often feels naturally qualified to make usability pronouncements. The oh-so-vital distinction, between being a "user" and a "representative user" tends to get lost. The experienced usability professional has had much practice trying to convince a product developer that the target audience for this product/site is likely not someone with a master's degree in computer science or electrical engineering, and is certainly not someone with 18 months of experience working on the project in question. The best way to drive this point home is to encourage software developers to observe usability testing (or participate directly in the usability evaluation; see, for example, Bias (1994)). Previously unable to comprehend that someone could not understand how to navigate a particular path through the software user interface, the product developer becomes convinced when he or she sees two or three test participants (the first one was just "stupid") repeat a problem. In lieu of having developers participate in or observe a usability test, making available video highlights of several normal-looking people encountering the same usability problem can be very convincing; drawing attention to such scenes by embedding media in usability-engineering reports can help to maintain a focus on actual users when interacting with other stakeholders.

Finding the Stakeholder Where the Cost and Benefit Lines Cross—What to Do When the Costs Outweigh the Benefits for a Particular Suborganization

Imagine the following scenario: armed with your usability cost-justification projections, you eagerly and confidently stride into a room of the decision makers at your software company. Having done your homework and generated an empirical, cost-benefit analysis, you propose a course of usability engineering that, while costing the development team an extra $20,000, will save a whopping $100,000 in the development of training materials. "Way to go," right? Well, if your audience consists of the company's director of product development and the director of training, they may well be underwhelmed by your proposal. Now, there is likely someone higher up the organizational structure who worries about both the development budget and the training budget, who will be more impressed, and more likely to be convinced by your cost-benefit analysis.

CONCLUSION

In case study after case study, usability is proving its worth. In a wide variety of domains using a wide variety of usability-engineering methods, those domains which spend money on professional usability engineering and take the time to calculate the ROI for those expenditures are finding that they are glad they did so.

- "An investment in user and task analysis early in the design and development process will reap considerable benefits in cost savings later in the process" (Hackos & Redish, 1998, p. 122).
- ". . . An executive development manager observed that reducing call volume by even 25% would cover the hiring of three additional support technicians" (Montaniz & Kissel, 1995).
- "Sun Microsystems has shown how spending $20,000 could yield a savings of $152 million" (Rhodes, 2000, p. 4).
- "Performing usability cost-benefit analyses within ones own company may be a first step toward introducing usability engineering techniques into a company" (Donahue, Weinschenk, & Nowicki, 1999).

- ". . . designers who have an understanding of financial issues and statistics will be better able to follow a standard ROI process, make the business case for their projects, design and specify and effective user experience, and, ultimately achieve the user and business goals" (Herman, 2004).

These quotations have offered encouragement and a plan for gathering cost and benefit projections, and actual calculations, and using a metric everyone can understand—dollars—to cost-justify a usability-engineering effort. We have offered some examples of spreadsheets to model this sort of cost-justification approach. Readers will have to derive their own values for the cost of a developer hour, the cost of a "basic" program of usability engineering, the likely reduction in customer support calls due to usability engineering, and so forth. Further, at no point mentioned earlier was the whole package put together. That is, if we pursue a certain program of usability engineering, what would be the grand total of all the benefits, including (but likely not limited to):

- Reduced costs due to development efficiency;
- Reduced costs due to fewer calls to the customer support line;
- Reduced costs due to less need for documentation;
- Reduced costs due to fewer training materials needed;
- Increased sales/visits;
- Increased return clients/visitors;
- Increased user throughput;
- Decreased customer-training costs;
- Decreased customer costs due to ability to hire less-sophisticated user audience;
- Reduced chance of lawsuits;
- Reduced chance of public relations disaster;
- Increased chance of trade-press praise;
- Increased customer satisfaction.

Our choice not to build one "grand total" cost-benefit analysis (showing, certainly, a ROI of hundreds of thousands of dollars to one) is intended to emphasize the importance of knowing the "value proposition" for your audience; the director of training may be unmoved by your argument to reduce training needs. (Indeed, this can be parried by arguing for those training dollars being spent on training the more sophisticated functionality of the product, and not the day-to-day tasks.)

Given the state of the software-development world today, though, usability engineering almost certainly will yield a robust ROI, in almost any UI/website development effort. By coming up with reasonable projections in advance, the usability advocate can gain the resources necessary to carry out that first program of usability engineering. By following up and calculating the "actuals" after the product ships or "goes live," the usability advocate will likely demonstrate how conservative the original projections were, and thus earn the confidence of the stakeholders for quite a long time. Thereafter, the cost-justification effort will become a way to continue to validate the usability expense, and to fine-tune the program of usability engineering, demonstrating which usability methods were the most cost effective for which types of products, at various points in the development cycle.

(Portions of this chapter appeared in a chapter by Deborah J. Mayhew and Randolph G. Bias in Ratner, J. (Ed.) (2003). *Human factors and web development.* (2nd ed.). Lawrence Erlbaum Associates.)

References

Bias, R. G. (1994). The pluralistic usability walkthrough: Coordinated empathies. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods.* New York: John Wiley & Sons, Inc.

Bias, R. G., & Keough, K. A. (2000). Usability triage for web sites. In the *Proceedings of the 6th Conference on Human Factors and the Web: Doing Business on the Web.* June, Austin, TX.

Conklin, P. F. (1996). Bringing usability effectively into product development. In M. Rudisill, C. Lewis, P. G. Polson, & T. D. McKay, (Eds.), *Human-computer interface design: Success stories, emerging methods, and real-world context.* San Francisco: Morgan Kaufmann Publishers, Inc.

Donahue, G. M., Weinschenk, S., & Nowicki, J. (1999). Usability is good business. Retrieved March 10, 2007, from http://interface.free.fr/Archives/Usability_Is_Good_Business.pdf

Forrester Research, Inc. (2000a). The best of retail site design, October.

Forrester Research, Inc. (2000b). Travel data overview. [White paper] August.

Forrester Research, Inc. (2001). Get ROI from design. [White paper] June.

Hackos, J. T., & Redish, J. C. (1998). *User and task analysis for interface design*. New York: Wiley Computer Publishing.

Herman, J. (2004). A process for creating the business case for user experience projects. In *CHI '04 extended abstracts on human factors in computing systems, Vienna, Austria*, 1413–1416. New York: ACM Press.

Karat, C. (2005). A business case approach to usability cost-justification for the web. In R. G. Bias, & D. J. Mayhew (Eds.), *Cost-justifying usability (2nd ed.): An update for the Internet age*. San Francisco: Morgan Kaufmann. (pp. 103–142).

Mayhew, D., & Bias, R.G. (2005). Cost-justifying usability for cross-cultural user interface design. In N. Aykin, (Ed.), *Usability and internationalization of information technology*. Mahwah, New Jersey: Lawrence Erlbaum.

Montaniz, F., & Kissel, G. V. (1995). Reversing the charges. *interactions, 2*(3), 29–33.ß

Nielsen, J. (2000). *Designing web usability: The practice of simplicity*. Indianapolis, IN: New Riders Publishing.

Rhodes, J. S. (2000). Usability can save your company. Retrieved March 20, 2007, from http://webword.com/moving/savecompany.html.

PERSPECTIVES ON HCI

FUTURE TRENDS IN HUMAN-COMPUTER INTERACTION

Andrew Sears
UMBC

Julie Jacko
Georgia Institute of Technology

What are HCI's Three Grand Challenges? **1282**
John M. Carroll 1282
Katsuhiko Ogawa 1283
P. L. Patrick Rau 1283
Gavriel Salvendy 1283
Constantine Stephanidis 1283
What Are the Three Most Important Results That Have Emerged from HCI Research in the last 10 years? **1284**
John M. Carroll 1284
Katsuhiko Ogawa 1284
P. L. Patrick Rau 1285
Gavriel Salvendy 1285
Constantine Stephanidis 1285
What Do You See as the Most Exciting Emerging Domains for HCI Researchers and Practitioners to Explore? **1285**
John M. Carroll 1285
Katsuhiko Ogawa 1286
P. L. Patrick Rau 1286
Gavriel Salvendy 1286
Constantine Stephanidis 1286
What Are the Most Innovative Changes You Envision in the Next Five Years with Regard to How People Will Interact with Information Technologies? **1287**
John M. Carroll 1287
Katsuhiko Ogawa 1287
P. L. Patrick Rau 1287
Gavriel Salvendy 1287
Constantine Stephanidis 1287
What Do Educators Need to Change to Ensure That PHD Students Are Prepared to Address The HCI's Grand Challenges? **1287**
John M. Carroll 1288
Katsuhiko Ogawa 1288
P. L. Patrick Rau 1288
Gavriel Salvendy 1288
Constantine Stephanidis 1288
What Is The Future Of HCI As A Discipline/profession? Will It Continue to be Sustainable as An Academic Endeavor That Yields Marketable Practitioners/Researchers? **1289**
John M. Carroll 1289
Katsuhiko Ogawa 1289
P. L. Patrick Rau 1289
Gavriel Salvendy 1289
Constantine Stephanidis 1289

Human-computer interaction (HCI) continues to evolve in terms of educational approaches, employment opportunities, and issues that are explored. After careful consideration, we decided to close the second edition of the *HCI Handbook* by posing a series of questions to five well-established members of the HCI community:

- John M. Carroll is the Edward Frymoyer Chaired Professor of Information Sciences and Technology and the Director of the Center for Human-Computer Interaction at Penn State University. His research interests include methods and theory in human-computer interaction, particularly as applied to networking tools for collaborative learning and problem solving, and the design of interactive information systems. He has written or edited 15 books, published more than 400 articles, serves on 12 editorial and advisory boards, and is editor-in-chief of the Association of Computing Machinery's (ACM) Transactions on Computer-Human Interactions. He has received both the Rigo Career Achievement Award and the CHI Lifetime Achievement Award from the ACM special interest groups on Documentation and Computer Human Interaction (SIGDOC and SIGCHI respectively), was elected to the CHI Academy, and received the Goldsmith Award from Institute of Electrical and Electronics Engineers (IEEE) and the Silver Core Award from the International Federation for Information Processing. He is a Fellow of the ACM, IEEE, and Human Factors and Ergonomics Society.
- P. L. Patrick Rau is a Professor of the Department of Industrial Engineering at Tsinghua University in Beijing where he founded and directs the Human-Computer Interaction and Usability Research Lab. His research and teaching activities have focused on human-computer interaction, mobile advertisement, design for aged people, and cross-cultural design. He has published articles on these topics in the HCI and related journals. He is a member of the Human Factors and Ergonomics Society, the Association for Computing Machinery, China Ergonomics Society, and Taiwan Ergonomics Society.
- Katsuhiko Ogawa was named the Director of Research Planning within the Department of Research and Development at Nippon Telegraph and Telephone Corporation (NTT) in 1995. He has served as the Director of NTT Cyber Solutions Laboratories since 2004. His areas of expertise include human interaction, media design, and home appliances. He has published more than 80 articles, is a member of Human Factors and Ergonomics Society, The Institute of Electronics, Information and Communication Engineers, and Information Processing Society of Japan.
- Gavriel Salvendy is a professor of Industrial Engineering at Purdue University and Chair Professor and Head of the Department of Industrial Engineering at Tsinghua University in Beijing. He is the author or coauthor of over 420 research publications and is the author or editor of 27 books. His research focuses on the human aspects of design, operation, and management of advanced engineering systems. He is the founding editor of the International Journal on Human-Computer Interaction and Human Factors and Ergonomics in Manufacturing. In 1995, he received an Honorary Doctorate from the Chinese Academy of Science. He is an honorary fellow and life member of the Ergonomics Society and fellow of Human Factors and Ergonomics Society, Institute of Industrial Engineers, and the American Psychological Association. In 1990, he was elected to the National Academy of Engineering.
- Professor Constantine Stephanidis is faculty at the Department of Computer Science—University of Crete. He is also the Director of the Institute of Computer Science—FORTH, and Head of its Human-Computer Interaction Laboratory and the Centre for Universal Access and Assistive Technologies. He is the President of the Board of Directors of the Science and Technology Park of Crete. His research interests lie in the fields of Human-Computer Interaction and Universal Access and many of his activities have focused on user interfaces self-adapting to user requirements and context of use. He has published more than 300 articles. He is the Editor-in-Chief of the Springer journal *Universal Access in the Information Society*. He is the founding chair of the ERCIM Working Group "User Interfaces for All" and of the International Conference "Universal Access in Human-Computer Interaction." He is actively involved with the European Commission's Research and Development Policy on Information Society Technologies with particular focus on issues related to e-Inclusion.

We began by asking about grand challenges in human-computer interaction (HCI). While others have asked about grand challenges in information systems, information technology, e-Science, and other specialized fields, we felt it was time to answer this question in the context of HCI. While the grand-challenges question looks toward the future, we also felt it was appropriate to reflect upon the accomplishments of the field. To do this, we asked about important results that have emerged from HCI research during the last decade. We concluded with a series of additional questions the future, with the goal of gaining insight into important application domains, interaction innovations, and the education of future HCI researchers who will address the grand challenges and produce future innovations. Lastly, we explored the future of HCI as a discipline and profession. While the length of the responses varies, we believe you will be able to identify common themes among the responses. We hope you will find useful suggestions to guide your research, ideas that may help shape educational programs, and ideas that will guide you in practice.

WHAT ARE HCI'S THREE GRAND CHALLENGES?

John M. Carroll

To broaden the question a bit, I think HCI faces different kinds of challenges. Since this interview is for a technical book, I assume you are most interested in engineering and scientific challenges. However, HCI also faces organizational/political challenges at several levels. These latter challenges are very important, since they embody threats to the whole enterprise of HCI. For example, consider the challenges of research funding. Nonmilitary funding and funding for basic research has significantly declined in recent years. Funding from industry, and even in-house industrial research funding, has significantly declined. National Sci-

ence Foundation research, and what research there still is in industry are more applied and more narrowly focused than in the past. None of this is good for HCI, and the solutions lie beyond anything technical theories and empirical results can provide.

The HCI community also faces many organizational challenges in constituting itself. There has been a persistent tension in product development organizations about how to organize usability functions. This is more complicated now because usability is not a single function and is frequently addressed by 3–4 different professional perspectives for example, user-experience analysis, graphic design, user interface and application development, and usability evaluation. All of these perspectives and professional skill sets are integral to HCI. This differentiation is a natural and very positive development; it shows again how important usability is to effective product development. But, now, instead of the simple battles between virtuous usability guys and lunkhead developers, we can have disputes between different kinds of HCI perspectives. It raises the question of how HCI is or will be a profession or discipline.

The engineering and scientific challenges are actually far less daunting, though a good deal more exciting. One is the emergence of ubiquitous computing—I prefer the European term "the disappearing computer"—as the reference paradigm for HCI. Of course, computing has been moving off the server, off desktop, and into personal devices for many years. And there is no shortage of attention to these developments, or interesting discussions of what they mean for research, development, and education in HCI. However, it is likely that we have only seen the very beginning of the diffusion of computing into the objects and experiences of everyday life. What will HCI become without the computer?

The second engineering and scientific challenge I would mention is "end-user programming." Ironically, some of the earliest work in what became HCI addressed the psychology of programming. In that work, early HCI researchers investigated how people understand and can use conditionals, how they can use Boolean logic to express queries, etc. This thread has never died completely, but at times it has appeared a somewhat arcane focus in HCI. I think it is more urgent than ever. As computing moves into the personal realm, it is essential for people to be able to exert more direct and wide-ranging control. It's not just nice to configure a web server, it's essential; indeed, it may soon be essential to have one running on your wristwatch. This challenge interacts with, and perhaps is compounded by, the disappearing computer: If manipulating explicit Booleans and other formal expressions in the relative calm of a desktop interaction is problematic, how will people get along drawing specifications in the air as they cross the street?

My third "grand challenge" is collaboration. I know we've had 20 years of computer-supported cooperative work (CSCW), and maybe 30 years of computer-mediated communication (CMC), but I think we still have only scratched the surface. One of the main functions of computers now is to give us access to e-mail, wikis, and blogs, so that we can interact with other people. The sophistication of collaborative systems is developing rapidly. Many project groups routinely employ collaborative infrastructures that a just few years ago would have been gems of the U.S. National Science Foundation's collaboratories initiative. A modern frontier is collaboration that never stops: Distributed global teams working in several time zones.

Katsuhiko Ogawa

In the next 10 years, we will likely see the emergence of three integrated services: (a) telecommunications and broadcasting, (b) fixed and mobile, and (c) telecommunication and information. The first challenge for HCI relates to the integration of telecommunication and broadcast services. In television, major changes in HCI will bring about TVs you can watch wherever and whenever you want, TVs with which you can watch broadcasts from all over the world via the Internet, and TVs where you can choose whatever picture you want from those being sent in. The next challenge will be HCI for mobile appliances. These will include cameras and music players, of course, and HCI for portable devices with an electronic money function will also become very important. Finally, there will be the challenge of HCI for cyberspace. Communication tools tailored to the individual will be designed to enable websites, communities, and chat rooms for the individual.

P. L. Patrick Rau

The first challenge is to make effective contributions in business, particularly in emerging markets such as China. In many companies, HCI people have successfully demonstrated their importance on product/service development. However, the contribution of HCI on the profit of a business is difficult to see. The second grand challenge is to develop and validate new methodologies. The pressure for improving HCI methodologies is increasing because of the rapid growth of Internet and telecommunication industries. The development lifecycle and the product lifecycle of handheld devices are decreasing. In addition, the importance of fun and emotional satisfaction for computers is growing, but current HCI methodologies are mostly designed for goal-oriented tasks. The final grand challenge is to develop new technologies that impact user experience. There will be some killer technologies that can change the way people use computing devices.

Gavriel Salvendy

The first challenge is to develop the science base of HCI. We also need to develop a comprehensive educational program for HCI, so that all computer-science and information-technology students would be exposed to HCI. Finally, HCI must to push the needed technology rather than respond to the emergence of technology.

Constantine Stephanidis

Computing technology evolves rapidly, and each new generation of technology offers new opportunities to improve the quality of human life. HCI is one of the key factors toward ensuring social acceptability of computer-based products and services and is therefore continuously called upon to face new challenges, by adapting, or better anticipating, new requirements for technology acceptance. Several recent trends in the direction of improving acceptability can be observed. Universal access aims at preventing the exclusion of users from the Information Society, while at the

same time increasing the quality and usability of products and services. Additionally, there are approaches to design that emphasise the need to better reflect users' everyday goals by providing more natural and activity-oriented, rather than application-oriented, forms of interaction, and by breaking down computing into specialized devices that are designed specifically for everyday activities. Other factors related to user acceptance include the need to provide self-explaining interfaces that limit the learning process to a minimum, as well as user interfaces that increase user trust and confidence. Another perspective is that of moving beyond usability and efficiency in design toward a more emotionally pleasant user experience. Consequently, a first grand challenge is that HCI must now explore new areas to understand and formalize the knowledge of what makes a design more acceptable, appealing, and fun to users, and to overcome the social, cultural, and organizational barriers that may limit acceptance of technological breakthroughs.

A second grand challenge, which is also related to the first, is the need to establish HCI theories and methodologies that are defined at a fundamental, more abstract level. HCI is a dynamic field, and the nature of interaction changes dramatically as time goes by. New technologies and trends emerge too often to be able to apply specific guidelines and practices to all of them in an equally successful manner. This is evident from the history of computing, which has started with command-line interfaces and has gone through a series of successive developments, from the graphical user interface, to mobile computing, virtual reality, and ubiquitous computing. In order to be prepared for new evolutions, a more fundamental approach to interaction needs to be elaborated. Human-centered design was a first attempt toward this direction. Although manifestations in design guidelines and interaction paradigms change along with technology, the basic principle remains the same: putting the human user at the center. Similarly, current approaches toward universal access transcend specific technological manifestations.

The third grand challenge concerns computer-based support and "digitization" of HCI practices. Today, most work in the design and evaluation of user interfaces is still being conducted "on paper." This is clearly becoming impracticable, as the task of HCI practitioners becomes increasingly complex due to the diversity of parameters involved and the need to design interactive environments that are entangled with the physical world; rather than simply designing user interfaces to single applications. In this respect, appropriate methods and tools are required.

WHAT ARE THE THREE MOST IMPORTANT RESULTS THAT HAVE EMERGED FROM HCI RESEARCH IN THE LAST 10 YEARS?

John M. Carroll

The last 10 years have been full of important results, factoring them into three bullets is the hard part!

One of the three would have to be *interactive-information visualization*. The past 10 years established the "spotfire" paradigm for allowing people to drive visualization toward the information they want. One classic example is the University of Maryland's real estate finder. The visualization is a map of the District of Columbia and its surrounds. A set of sliders is provided, allowing the user to specify a price range, as well as indicating locations to which they will commute or frequently travel to, and how far from these locations, they are willing to live. As the user tunes these sliders, spots appear/disappear on the map indicating currently available homes that meet the current search specification. This project utterly transformed how people think about information visualization. The basic direction is now being most dramatically pioneered in virtual environment work.

My second candidate would be *collaboration via the Web*. The original web itself is barely 10 years old, and was a huge HCI result: usable ftp (file transfer protocol). The important lesson I see in this is how a relatively modest enhancement in functionality can trigger a revolution in the use of technology. Subsequent refinements to the web, like forums, wikis, and weblogs are even more important than the original web. New web technologies dissolve the old distinction between synchronous and asynchronous collaboration. They have made web-based collaboration so easy, so flexible, and so powerful that—like e-mail—we now use it just to work with folks in the next office. Anyone can now create and manage a sophisticated collaboratory environment. It is difficult to imagine managing a project without these tools.

Thinking more broadly about social interaction, many of these same tools for web-based collaboration have changed the fabric of society. Sixty million people are cultivating weak ties on myspace.com, while 30 million are developing strong ties on facebook.com. This is a nice validation for Mark Granovetter, who originated the distinction between relatively more intimate (strong) ties to those we regularly interact with, and relatively more casual (weak) ties to those we merely know, but who can provide specific and critical help in specific areas, such as advice in a job search.

I'm starting to hate this "top three" thing! My third important result is *powerful information retrieval tools*. This is a bit of a cop-out, since I am using it to aggregate a whole bunch of now-paradigmatic mechanisms that are quite distinct, but generally help us find stuff—latent semantic analysis (LSA), which represents information structures by patterns of co-occurrence and social recommendation, as employed by Amazon, which allows us to make use of the prior searches of ourselves and others in streamlining current searches, and Google's search algorithms, which make searching the entire web easier and more successful than looking for information on your desk.

Katsuhiko Ogawa

One important result is universal designs for aged and disabled persons. Another is interaction between portable devices and users, such as that in portable phones, portable computers, iPods, and digital cameras. And another is interfaces for the dispatching of individual information, such as blogs and homepages. To be sure, these did not surface directly from HCI research, but it is certainly true that the merits and advantages of HCI were the deciding factors in enabling these to become widely available.

P. L. Patrick Rau

Website usability is clearly one of the most important advances in the last decade. Many HCI researchers have been working on improving user experience on the web. I would also include advances and significant progress in the area of user interfaces for handheld devices. Handheld devices such as cell phones and MP3 players have drawn attention of many HCI researchers.

Gavriel Salvendy

The first would have to be the broad area of concepts, methods, and tools, which have been developed to accommodate all segments of society. I would also include developments and methods relating to visualization and the development of adaptive-interface methods.

Constantine Stephanidis

The user-centered approach to design is probably one of the most important benefits arising from HCI research in the last decade. Moving from a technological perspective of design to placing the user as the first priority has fundamentally changed the way we design artifacts in all domains. In the computing world, we have seen the impact of this process most prominently in user interfaces, ranging from software applications and services to operating systems and web pages. The result has been an increase in the usability of computers and the quality of users' experiences. Furthermore, having the user at the center of the design is a perspective that is most likely to encourage the advancement of HCI as a discipline. For example, there have been steps forward in understanding the complexities of human cognition, emotions, and needs because the focus of investigation has shifted from technology to the human being: Pervasive Computing seeks to hide the technology; Emotional or Experience Design seeks to make interactive products not only more efficient but also more pleasant, and Universal Access aims to provide the means for all humans to benefit from information technologies. A related result has been the development of methods to evaluate the usability (and hence efficiency and quality) of designs and products. The past decade has seen considerable research in developing reliable and substantial ways to determine if a product suits the real requirements of users (as opposed to the perceived requirements). Appropriate heuristics and empirical evaluation methods conducted in the laboratory have started to become the norm in the industry, although there is still much room for improvement in this respect. Work on usability has also been consolidated in a wide range of standards that bring design knowledge nearer to wider practice.

Recently, considerable advances have also taken place in the domain of computer accessibility, which is now recognized as an important prerequisite for technology acceptance and a fundamental aspect of universal access. There is now a considerable body of knowledge on how to make interactive applications and services accessible by people with various types of disabilities, but also of different ages and cultural backgrounds. A typical example of concrete advance in this domain is that of web accessibility, for which there are now universally recognized de facto standards, and, in many countries, also regulatory frameworks.

Another recent advance in HCI research, where considerable results have been obtained, although many issues remain to be explored, concerns user interface personalization. Adaptation methods and techniques have been investigated, and new forms of interaction have emerged through multimodality and multidimensionality of user interfaces. This area of research is one of the most exciting, as the adaptation of interaction depending on the context of interaction is crucial for universal access, not only for disabled users but also for the population at large. For example, nonvisual interaction is more generally useful beyond the case of blind users.

WHAT DO YOU SEE AS THE MOST EXCITING EMERGING DOMAINS FOR HCI RESEARCHERS AND PRACTITIONERS TO EXPLORE?

John M. Carroll

To me, the term "domain" is a bit problematic, but in an interesting way. System architectures define domains. Types of functions define domains. Applications define domains. And user experiences define domains. All these different senses of the term suggest exciting emerging domains for HCI.

New kinds of systems always entrain exciting emerging domains for HCI. The system infrastructures for the disappearing computer are obviously creating new domains for HCI in mobile interaction and distributed agency. One of the most important of these system-driven domains is security and privacy. In the mainframe and client-server worlds, these are central-system administration services. However, increasingly, they are end-user concerns. Security and privacy is already an important HCI domain, and will be far more important in coming years.

New types of functions also define emerging domains for HCI. One important area of this sort is universal accessibility. Not everyone wants or would tolerate a user interface with extra-large text, but for some users, the possibility of large fonts makes the difference between having access to online services and not having that possibility. Identifying the critical functions to modulate user interface presentations and controls to better reflect the diversity of human competencies and preferences is a major ongoing focus in HCI. For example, there was a record-setting response to a recent special issue of the *ACM Transactions on Computer-Human Interaction* on accessibility. Another example of a function-driven HCI domain is web authoring. Wiki functions for linking, posting, and discussing on the web have enabled social computing to leap beyond the limitations of chat and instant messaging. Now users can have something to talk about, and can collectively build spaces and information structures. Like the original web itself, this shows how modest enhancements in functionality can trigger explosions in the appropriation of technology by people.

New application domains have always been a key driver for HCI, and this is still the case. For example, health care—medical

and health informatics—is a nexus of incredible need and incredible opportunity. Much of it is quite mundane technologically, but the potential social impacts are huge. This is probably the center court for HCI applications in the next decade, but there are lots of others. Community informatics is one that I have worked in over the past decade or so: The public and civic sector, particularly at regional and local levels, is literally where we live, but impoverished with respect to information technology, in generally, and HCI in particular. There are wonderful and inspiring cases of initiatives, some of which have also been very successful as public relations projects, but the broad state-of-the-art in regional and local community informatics is still fairly primitive. I believe that the rapidly growing demographic of educated and engaged elderly is going to be a key to a revolution in community informatics during the next decade.

The most exciting emerging domain with respect to user experience is affect. HCI started out worrying about whether user interfaces experienced as "easy to use" could also be experienced as "easy to learn," or whether there might be some sort of terrible tradeoff that would force designers to choose. Things turned out to be much more interesting than that! Now, when we speak of the experience of "usability," we are usually referring to an ever-growing complex of attributes: easy to use, easy to learn, challenging but tractable, consistent with prior knowledge, supportive of existing workplace roles and practices, just plain fun, and evocative of feeling of well being, creativity, collective efficacy, cultural identity, and social capital. I doubt that this is the end of the list.

Katsuhiko Ogawa

Portable devices that are easy for aged and disabled persons to use and services allowing access to information from the Internet using these devices. Search functions that allow users to find beneficial information are also very important.

P. L. Patrick Rau

My list would include emotional design, HCI in computer games, HCI in smart environment, and cross-cultural design. Design for fun is the design concept of many consumer products/services. Functions are no longer the only focus because of the maturity of some information technologies. Bringing more fun in use means value to business. Playing computer games has become a common life experience for many younger people. Many topics can be found about HCI in computer games. Customization is getting easier, which reduces the cost of addressing the special needs for different cultures.

Gavriel Salvendy

The first area I would list is the development of HCI theories and methods to enable the effective use of nanotechnology based computers. I would also include developing a comprehensive understanding of how individuals in different cultures make decisions; how the same input results in different decision outcomes across cultures; and how using this know-how to develop an automatic translation across cultures (not languages). Finally, I believe that HCI will emerge as a system science, whereby HCI will be studied as part of the overall job of the individual rather than as an isolated entity.

Constantine Stephanidis

The Information Society is anticipated to evolve toward the proliferation of computational systems that integrate a range of networked interactive devices embedded into a physical context (either indoor or outdoor spaces) to provide a host for a broad range of computer-mediated human activities and access to a multitude of general or specific services and applications. Such systems assume that computers and networks are distributed around physical environments, and are anticipated to exhibit intelligent and context-sensitive behavior. Although currently it is not yet clear how the vision for ambient intelligence is to be realized and shaped, its evolution is conceivable because of some general development trends that have already started to materialize. For example, services will be dynamic and can be reconfigured or recombined at run time to accommodate the different needs of different users in different environments. There will be no clear distinction between interpersonal communication and access to information; different components that use different media will be interconnected to allow a free intermixing of these functions. Services will be highly interactive; for example, interaction will be complex in terms of the functionality offered, input required, output provided, dialogue structure, and configuration capabilities. Most services will be inherently multimedia, providing information in multiple types of media (e.g., sound, graphics, video, text, animation, etc.) simultaneously and in an integrated manner. Additionally, interaction will often be multimodal; for example, different sensorial and motor abilities will be concurrently used, consciously or unconsciously, and based on more natural forms of dialogue. Cooperation will be an important aspect, communication and access to information will be concurrently used to solve common problems in a cooperative manner; moreover cooperation may be among human users themselves or among user representatives (agents and avatars), to whom variable degrees of trust can be assigned. Finally, computing will become progressively more social. Access to information and communications will no longer be the task of an individual but will be extended to communities of users, who have at their disposal common (sometimes virtual) spaces in which they can interact. This paradigm shift offers exciting opportunities for HCI research, as technologies are anticipated to have a much bigger impact on human life in the future, and raises an entire series of issues to be addressed, including interaction implicitness and continuity, distribution of input and output, appropriate exploitation of human perception and cognition abilities, user control over the intelligent environment, health, safety, privacy and social issues, and dynamic context awareness. It is clearly very important that such an environment develops proactively in a user-centered fashion, and comes to satisfy people's needs and improves quality of life, avoiding the risk of a technological nightmare. It is also very likely that new methods, techniques, and tools will be necessary for supporting the user interface development lifecycle.

In addition to human and computing devices, the intelligent environment will also be populated by robots. As a consequence, human-robot interaction will constitute a relevant domain of investigation, opening exciting new perspectives and offering significant potential improvements of the quality of life. In this context, the robot behavior and its autonomy with respect to the human user are important factors of acceptability.

Finally, we are witnessing a gradual fusion of computing with communications, learning, and entertainment. This also calls for new HCI approaches addressing appropriate design approaches and techniques, usability studies, theoretical and pedagogical foundations, analysis of nonbusiness human activities, as well as social and cultural issues.

WHAT ARE THE MOST INNOVATIVE CHANGES YOU ENVISION IN THE NEXT FIVE YEARS WITH REGARD TO HOW PEOPLE WILL INTERACT WITH INFORMATION TECHNOLOGIES?

John M. Carroll

My responses to the first three questions have already addressed this to an extent, probably to an extent just beyond my actual competence! My crystal ball is a lot like everyone else's, I suppose. The descendents of today's cell phones and other wireless devices will be far more powerful, more flexible, and more usefully connected to databases, to their own current locations in the world, and to the people we care about, work with, etc. How could it be otherwise?

I think we can look forward to some clarifications in the crystal ball during this period. For example, during the past five years, there has been debate about the feasibility and desirability of the agent paradigm. I'm not in the middle of it, but to a bystander, it echoes perennial debates about artificial-intelligence (AI) applications. I think it is likely that this latest instance of the AI debate will be resolved within the next five years; we'll either have pervasively integrated agents in our information interactions, or the agent paradigm will be gone, and AI will have made another strategic retreat to something analogous to expert systems. This is probably controversial enough to get me into big trouble!

Katsuhiko Ogawa

Agents will take the place of robots with moveable arms and legs. Agents will be incorporated into portable devices, and people can have their own agents incorporated into net space. These agents act as proxies for people, amassing information and making it available to others.

P. L. Patrick Rau

People will have wearable computers and ubiquitous computing. They will be able to access to information anywhere at any time. User interface of computers no longer means a display plus a keyboard and a mouse. Wireless network will bring information into people's pockets.

Gavriel Salvendy

My list would include additional emphasis on the disappearing computer, the emergence of miniaturized computing systems, and the revitalization of intelligent interfaces.

Constantine Stephanidis

There is a considerable trend toward mobile interaction. Mobile technologies are everywhere. Cell phones in particular have become a modern-day need and manufacturers have started to expand their capabilities. Like all new technological innovations, many areas are still in need of improvement, especially since cell phones have started to evolve into multifunctioning equipment, combining features such as personal organizers, web browsers, cameras, e-mail clients, and music players along with traditional telephone facilities. The main issue regarding mobile technologies is that users have already been using these applications in a very different environment and have not yet adapted to the cell-phone device. Right now, it is a race of arms between manufacturers to be the first to include new innovations to their devices and not spending much time designing the existing ones to best do their job. The same holds true for PDA devices; although they seemed to have caught on well in in the business world, they did not manage to penetrate the general consumer market. The Information Society is not going to be confined in an office or at home. The convergence of communications and computing, in the form of cell phones, PDAs, and notebooks connected on a wireless network, means that quality experience on the move is going to be in high demand in the very near future.

A similar trend toward networked interaction can also be observed in the home environment, where technology is mature enough to support, for example, the fusion of the web and the digital TV, as well as the overall "connectivity" of various devices and controls in the home environment. As a consequence, we are already witnessing, for example, web navigation and interaction through a remote control, or using a PDA to set the fridge temperature.

Another field that is very likely to yield interesting applications in the near future concerns the exploitation of biometrics for interaction purposes, mainly for verification and identification purposes, but also, perhaps in the longer term, for the recognition of cognitive and emotional states.

WHAT DO EDUCATORS NEED TO CHANGE TO ENSURE THAT PHD STUDENTS ARE PREPARED TO ADDRESS THE HCI'S GRAND CHALLENGES?

John M. Carroll

The most important thing we need to change is our disciplinary bias. No one, not even the professors in our top HCI programs, can be masters of everything that is HCI. Experts are always narrow in some sense. HCI has matured as an academic focus in several disciplines—arts and architecture, cognitive science, communications, design, education, geography, information

science, industrial engineering, library science, and psychology. Like the blind men and the elephant, each of these has its own limited view of what HCI is, and produces students with these limited views perfected.

The obvious way to address this is for faculty and students to cross boundaries and expose their narrow expertise to broader discourses. Of course, this already happens in some places, but I think it is still talked about more than practiced. I guess one thing I would say to students is that if you think you are getting this narrow sort of training, don't settle for it!

The second thing I think we should change is our emphasis on practical skills. I love John Dewey; reading Dewey changed my life. However, I think that many pedagogical implementations of Dewey's ideas are too simplistic. The mere fact that students carry out "authentic tasks" in their classes is not a guarantee of anything, and in fact, can easily be gamed. One example of this is role definitions in project teams. A common approach is to create teams with a project manager, a lead designer, an information designer, and an implementer. However, these roles have only modest operational dependencies (which of course is why they are used in real work). Thus, they actually provide students with a gift-wrapped approach to divide-and-conquer; that is, to minimize overall group effort, which incidentally minimizes overall learning. Moreover, students sometimes specialize in these roles (which optimizes effort with respect to grading outcomes). After 5–6 courses in which a given student has repeatedly played the role of, say, information designer, how much has he or she learned about implementation?

I think that more attention has to be given learning objectives and outcomes, and not just to apparent authenticity. Another aspect of this, I think, is the erosion of emphasis on theory. HCI is a cornucopia of theory: Theories at different levels of philosophical, behavioral, social, cultural, economic, and technological analysis. Training students in usability testing or in the use of current development toolkits is appropriate, but not adequate. Such narrow skills will quickly become obsolete, they probably already are. Again, I would suggest to students: If you have learned how to write an interaction scenario, create a running prototype, and run a usability test, but you didn't learn about human-information processing, mental models, activity theory, small-group behavior, and ethnomethodology, you deserve a refund.

Katsuhiko Ogawa

It is important for them to learn not how to evaluate and analyze tasks, devices, software and so on, but to actively challenge the task of designing new and innovative things. They should also learn how to design new services as well as new things.

P. L. Patrick Rau

PhD programs should stress issues related to ubiquitous computing while providing basic knowledge to guide future research and development. Programs should also introduce new technologies, such as nanotechnology and cognitive science. Finally, HCI students need to understand developments in different areas. The development of HCI in Asia like Japan, Korea, Taiwan, and China will benefit the global HCI community by either supplement investigations of Asians in previous HCI areas or inspiring new research issues from a different cultural perspective.

Gavriel Salvendy

PhD programs need to emphasize the science base of HCI and provide courses that prepare students with the know how on the science of the discipline.

Constantine Stephanidis

The foremost challenge in HCI education is addressing multidisciplinarity. A variety of diverse disciplines contribute to HCI, including ergonomics and human factors, social and organizational psychology, cognitive psychology, software engineering, design, artificial intelligence, semiotics and linguistics, philosophy, sociology, computer science, and anthropology. The leading experts of HCI come from all sorts of backgrounds, and the ability to widen scientific interests to address different fields is a key toward advancing the understanding of how humans interact with computers. An HCI expert must have solid multidisciplinary roots; he or she must be a scholar in more than one field and, more crucially, be able to merge the knowledge from each area into a heterogonous science. One of the problems today in HCI is that although everybody understands the multidisciplinary nature of the field, few have the level of in-depth knowledge that is required to push the boundaries further. Sometimes people work on preconceptions of what each discipline claims, and seldom take into account the most recent advances in each discipline. It is, therefore, necessary to adopt a more holistic approach to education, address multidisciplinarity more effectively in university curricula, and encourage students to cultivate a multidisciplinary mentality. The ultimate goal is that HCI researchers are able to evolve with, and anticipate, new and emerging technological environments and societal needs. Toward this direction, current efforts focus on examining educational curricula and practices from across the world, exploring the topics that are taught and the techniques used, discussing the foundations of HCI, examining how the curriculum is changing and the driving forces for this change, reviewing how students exploit their HCI education, and investigating employers' requirements for HCI curricula. Also from the perspective of HCI education, HCI theory is of particular importance, as it is an instrument to equip students with a "durable" background in a very dynamic environment.

WHAT IS THE FUTURE OF HCI AS A DISCIPLINE/PROFESSION? WILL IT CONTINUE TO BE SUSTAINABLE AS AN ACADEMIC ENDEAVOR THAT YIELDS MARKETABLE PRACTITIONERS/RESEARCHERS?

John M. Carroll

I'm glad you asked that question! I have changed my mind on this recently. Until a few years ago, I felt that future of HCI be-

longed to a new kind of technical professional—computer scientists with significant understanding of usability and design. I envisioned HCI as a major new focus within the discipline of computer science. I spent 10 years as a professor of computer science—five as a department head—trying to realize this vision in a sort of "best-case" situation with respect to like-minded colleagues and institutional support.

I changed my mind because I came to see that there is far too much to know and do in HCI professional activity for one-size-fits-all scenario. As I mentioned in an earlier answer, HCI is a constellation of professional roles, skills sets, and perspectives. This is a good thing, and it will surely provide richer and more meaningful careers for HCI professionals! However, it eliminates my earlier vision of a sort of super professional. Indeed, looking at contemporary trends in computer science and other computing-related disciplines suggests that the trajectory of HCI within computer science is destined to become narrower, not broader. As a former computer-science professor, this makes me sad, but it is clearly an opportunity for many other disciplines.

HCI academic programs are developing in arts and architecture, cognitive science, communications, design, education, geography, industrial engineering, library science, and psychology. One of the most important recent developments is the emergence of "i-schools"—schools of informatics, information studies, and information sciences and technology—as academic homes for HCI programs. The core construct for i-schools is information, not computation, as is the case for computer science. HCI ends up in the center of i-schools, instead of at the periphery of computer science. When I realized that the role of HCI within computer science was narrowing instead of growing, I moved to an i-school.

I think that HCI has a strong future both as an academic endeavor with respect to research and education and as a professional focus. As I look back over my 30 years in HCI, it seems that at every point the opportunities far outstripped the people power available, and the future looked more so. This is exactly how it looks to me today—an abundance of opportunities for contributions from HCI. Because the endeavor is increasingly multidisciplinary, I do see a need for creative constructivism in articulating a big vision that embraces and fully utilizes the diverse professional roles, skills sets, and disciplinary perspectives that constitute HCI. For example, we should start talking more about multidisciplinary teaching in HCI and about multidisciplinary HCI teams in product development. The greatest risk I see is not that HCI is going to flop or fade away, but that it is going to fracture into several pieces—each of which would be quite exciting and important in itself.

Katsuhiko Ogawa

If, like a doctor, we have the clear goal of curing disease and sickness, it is critical for us to always have the most up-to-date knowledge available when treating patients. As with doctors treating their patients, with HCI, too, in giant factories, aircraft, automobiles and the like, safety should be the first priority. In the area of consumer products, however, first priority is often given to the designing of functions and operations. Consequently, we should move away from the academic aspects stressed in the HCI of today and concentrate more on the practical aspects.

P. L. Patrick Rau

HCI will grow as the information technology (IT) industry grows. In China, the IT industry is now undergoing a rapid growth which North America and Europe experienced. This provides opportunities for HCI development. Especially after 2004, HCI has gained a lot of attention from the industry in China. However, there are also challenges. The development of such an interdisciplinary field such as HCI in academia will take more time and effort and will be strongly influenced by the industry.

Gavriel Salvendy

I view HCI in computer science as analogous to differential equations in mathematics. In the early stages, mathematicians went around engineering departments to help engineers solve their problems using differential equations. Over the years, differential equations became an integral part of engineering education and all engineers were acquainted with differential equations and no longer needed the mathematicians. Similarly, computer scientists and information-technology specialists will all be trained in HCI, and the demand for practicing HCI will significantly decrease over the years.

Constantine Stephanidis

HCI helps in the design and development of better products; i.e., more useful products that do what the users want them to do easily, efficiently and, in the best case, pleasantly. From the application of HCI methodologies throughout the development lifecycle, from analysis to design guidelines and practices to development strategies, and finally to evaluation and testing, there is clearly a major role that HCI plays toward improving interactive products and services. Evaluation of usability is perhaps the area of HCI that prevails, because it is permanently embedded in the business world. However, there is a clear need for wider HCI practice in industry, as only the major players have invested heavily in such direction, with clearly positive results, both product-wise and business-wise.

Thus, there is an ever-increasing need for HCI practitioners and researchers. However, in order to develop professionals to fulfil this need, enhance the practice of HCI in industry, and facilitate technology transfer, HCI teaching in universities should be rethought. First of all, the next generation of practitioners needs to be prepared for the business world, for example, by learning basic management and marketing skills. Additionally, HCI practitioners need to be better prepared to address the evolving situation in the context of the Information Society. A typical example in this respect is universal access and design for all. Although there is now wide recognition and a consolidated body of knowledge, one of the critical impediments to the adoption of universal design in practice is the lack of qualified practitioners

who understand diversity in the user-target population and in the technology and in the context of use and are able to integrate the related requirements in the development process. To overcome this, it is recommended that, in the short term, accessibility, usability, and quality in use are introduced as mandatory components of university education. Other similar challenges are very likely to emerge with the anticipated evolution of the information society toward an ambient-intelligence environment, where the complexity intrinsic in developing interactive applications and services will grow even further. As a consequence, HCI experts must be equipped to address diverse technological environments and changing needs in a dynamic way. It is likely that, in the future, "specializations" of HCI expertise in different subfields will arise.

AUTHOR INDEX

Page references followed by *f* indicate figure.
Page references followed by *t* indicate table.
Page references followed by *n* indicate footnote.

A

Aarås, A., 218, *225*
Abaluck, B., 667, *677*
Abascal, J., 560, *571*
Abbate, J., 547, *554*
Abbott, K. R., 551, *554*
Abbott, T. S., 725, *737*
Abdelhamid, I. S., 765, *773*
Abelson, H., 804, *807*
Abelson, R., 818, *828*
Aberg, J., 445, *455*, 1254, *1261*
Abley, M., 490, *505*
Abotel, K., 1205, *1207*
Abowd, G., 171, 172, 269, *292*, 854, *867*
Abowd, G. A., 192, 193
Abowd, G. D., 288, *292*, 552, *554*, 642, *656*, 1251, 1253, 1255, *1261*, *1263*
Abraham, L. J., 1226, *1239*
Abrams, M., 1123, *1124*
Abrams, R. A., 29, 31, 33, 34, 35, *39*, *40*, 50, *60*
Abry, C., 423, *428*
Accot, J., 162, 166, 166*f*, *172*
Acero, A., 862, 863, *868*
Ackerman, M. S., 501, *505*, 552, *554*, 642, 643, 647, *655*, 1251, *1261*
Ackerman, P. L., 122, *130*
Ackoff, R. L., 6, *22*, 1044, *1057*
Adam, S., 51, *59*
Adamczyk, P. D., 439, *456*
Adami, A., 1228, *1238*
Adamic, L. A., 550, *555*
Adamis, A. P., 856, *868*
Adams, A., 642, 645, 647, 654, *655*
Adams, B., 796, *808*
Adams, E., 495, *506*
Adams, J. A., 28, *39*
Adams, K., 842, 849, *851*
Adams, N. I., 171, *176*
Adams, W. G., 666, *676*
Adcock, J., 863, *868*
Adeney, D., 1257, *1266*
Adesanmi, D., 1250, *1264*
Adjouadi, M., 841, *849*
Adjoudani, A., 417, 420, *428*
Adler, A., 1245, *1264*
Adler, P. S., 1244, *1261*
Adler, R., 780, *790*
Adler, T. F., 771, *774*
Adlin, T., 18, *24*, *926*, 941, 993, 1015, *1016*, 1131, *1149*
Agar, M., 969, 973, *986*
Agarwal, R., 901, 912, *915*
Agarwala, A., 474, *485*
Aggarwal, V., 443, 444*n*, *458*
Agrawala, M., 170, *174*
Agre, P. E., 1247, 1251, 1253, *1261*
Ahamad, M., 650, *656*
Ahlberg, C., 497, *505*, 510, *542*
Ahrenberg, L., *390*, 420, *429*
Aida, H., 863, *868*
Aiken, A., 646, *658*
Aiken, Lewis R., 1160, *1168*
Ainsworth, K., 612, *618*
Ainsworth, L. K., 575, *599*, 929, *947*, 1199, 1202, 1205, 1206, *1207*
Airasian, P., 803, *807*
Aish, R., 474, *485*
Aizawa, A., 566, *571*
Akamatsu, M., 168, *172*, 624, *635*
Akamatsu, S., 85, *90*
Akoumianakis, D., 859, *870*, 1101, *1105*
Akrich, M., 1243, *1261*
Aksit, M., 1115, *1125*
Al-Saleh, O. I., 705, *718*
Albacete, P. L., 840, *849*, *866*
Albers, M., 318, *324*
Alborzi, H., 797, 798, 799, *806*, 1065, 1069, 1070, *1078*
Aldrich, F., 394, *411*, 798, *808*
Alessi, S. M., 138, *144*
Alexander, C. A., 491, 493, *505*
Alexander, I. F., 1054, 1055, *1057*
Alexander, K. J., 796, 804, *807*, *809*, 1143, *1148*

Alexandersson, J., 414, *431*
Alfaro, L., 217, *225*
Algazi, V. R., 624, *635*
Ali, S. Y., 705, *718*
Allaire, A. J., 785, *790*
Allan, R. W., 1141, 1147
Allanson, J., 270, 277, *293*, 841, *849*
Allbeck, J. M., 626, *635*
Allen, J., 219, *226*
Allen, R. B., 725, *737*
Allen, T. W., 70, *74*
Alleva, F., 862, *868*
Allgayer, J., 414, *430*
Allgood, C. F., 1056, *1059*
Allport, A., 33, *39*
Ally, M., 610, *618*
Alm, N., 817, 818, 825, *827*, *828*, 838, 840, *849*, 872, *882*
Almor, A., 422, *428*
Alpert, S. A., 1047, *1057*
Alpert, S. R., 1054, *1058*
Als, B. S., 799, *806*
Alshuth, E., 218, *226*
Altman, I., 642, *655*
Altmann, E. M., 100, 101, *111*
Altshuld, J. S., 575, *602*
Alty, J. L., 249, *263*, 394, 400, *410*
Amalberti, R., 574, *597*
Ames, S., 647, *655*
Amir, A., 417, *430*
Amory, C., 970, *987*
Anagnost, T. D., 1249, *1264*
Anagnostou, G., 474, *485*
Anastasi, R., 285, *292*
Andelman, D. A., 902, *915*
Anderson, D., 474, *485*
Anderson, H., 967, *987*
Anderson, J. R., 56, 58, *58*, 96, 99, 106, 107, *111*, *112*, 168, *172*, 217, *226*, 445, *455*, 575, *597*, 634, *635*, 804, *807*, 1196, *1206*
Anderson, K. L., 492, *507*, 986*n*, *986*
Anderson, L. R., 1022, *1038*
Anderson, P., 633, *635*
Anderson, R., 646, 649, *655*, *658*
Anderson, R. B., 68, 69, 70, *75*
Anderson, R. E., 1257, *1261*
Anderson, R. H., 547, *555*
Anderson, R. J., 967, 986, *986*, *987*
Anderson, S., 874, *881*
Anderson, T., 608, 610, *618*
Anderson, W. L., 1075, *1077*
Andersson, B. J. G., 319, 320, *324*
Andes, R. C., 583, *597*
Andrasik, F., 1226, *1238*
Andre, T. S., 1141, *1147*, 1177, 1178, 1185, 1186, *1188*
Andreasen, P. N., 819, *827*
Andres, C., 912, *915*
Andrews, C., 715, 715*t*, *717*
Andrews, J. W., 735, *740*
Andrews, P. W., 578, *598*
Andriessen, J., 596, *598*
Androutsellis-Theotokis, S., 283, *292*
Ang, C. S., 614, *618*
Angel, E. S., 151, *157*
Angell, L., 708, *718*
Angelo, J., 835, 843, 847, *851*
Anhalt, J., 300, *311*
Ann, E., 939, *946*
Annett, J., 150, *157*, 575, *597*, 1199, *1206*
Anson, J. G., 33, *41*
Antón, A., 1054, *1057*
Antonietti, A., 765, *774*
Antoniou, A. A., 83, *91*
Anzaldúa, G., 1064, *1077*
Aoki, H., 846, *850*
Apacible, J., 439, *456*
Apitz, G., 165, *172*
Appadurai, A., 1124, *1261*
Apperley, M. D., 537, 538, *543*
Applegate, W. B., 858, *866*
Apter, M. J., 84, *92*
Aquirre, A., 873, 876, *883*
Arbanas, J., 879, *882*
Arce, R., 765, *774*
Archer, W., 610, *618*
Arditi, A., 854, 858, 865, 866, *866*, *867*
Arehart, C., 280, *292*
Arens, Y., 399, *410*
Argüelles, S., 782, *790*
Argyle, M., 424, *428*, 549, *555*
Arias, E., 475, *486*
Arida, S., 474, *487*
Arnall, T., 904, *915*
Arnold, A., 859, *870*
Arnold, P., 474, *486*
Arnott, J. L., 86, *91*, 817, 818, *827*, *828*, 840, *850*
Arnowitz, J., 1018, *1038*
Arocha, J. F., 674, *677*
Aron, K., 726, *738*
Arons, B., 169, *172*, 262, *263*, 278, *292*, *294*
Aronson, J., 574, *602*, 766, 770, 773, *773*, *775*, 886, *892*
Aronsson, L., 606, *618*
Arrow, H., 547, *555*, 583, *599*
Arthur, W. C., 733, *737*
Arvo, J., 626, *636*
Arvola, M., 1015, *1015*
Arvonen, T., 1048, *1058*
Aryal, B., 713, *719*
Åsand, H.-R. H., 1067, *1080*
Asanuma, N., 233, *244*
Asch, D., 873, 876, *883*
Ashbourn, J., 650, *655*
Ashbrook, D., 310, *312*
Ashman, H. L., 560, 561, 564, 565, 567, 568, *571*, *572*
Ashok, M., 861, *868*
Ashworth, C. A., 1066, *1080*
Asimov, I., 8, *22*
Aspinwall, L. G., 121, *130*
Athenes, S., 29, *40*, 259, *264*
Atkeson, C. G., 182, *196*, *198*, 1255, *1262*
Atkins, D. E., 547, *555*, *557*
Atkins, D. L., 643, *656*
Atkinson, M. L., 590, 592, *597*
Atkinson, M. P., 1173, 1174, 1178, 1180, 1181, *1189*
Attewell, P., 1248, *1261*
Attneave, F., 492, *505*
Atwell, P., 764, *773*
Atwood, M. E., 12, *23*, 56, *58*, 95, *111*, 286, *293*, 490, *506*, 929, *947*, 1053, *1058*, 1141, *1147*, 1179, *1188*, 1198, 1199, *1207*
Aung, H. H., 845, *852*

Aunon, J. I., 183, *197*, 841, *851*
Austin, B. T., 663, *678*
Austria, A., 708, *718*
Avendano, C., 624, *635*
Averill, J. R., 79, *89*
Avon, R., 764, *774*
Avons, S. E., 46, *59*
Avrahami, D., 1255, *1262*
Axelsson, A.-S., 583, *601*
Ayatsuka, Y., 171, *175*, 182, *198*
Aye, T., 204, *225*
Ayers P., 397, *410*
Aykin, N., 361, 362, 363, *377*
Ayoub, M. A., 724, *739*
Azarbayejani, A., 180, *198*
Aziz, M., 57, *58*
Azuma, R. T., 204, 223, *225*, 299, *311*

B

Baauw, E., 801, *807*
Babbie, E., 972*n*, 986
Baber, C., 148, 150, 151, 152, 153, 154, 155, *157*, *158*
Baca, J., 862, *867*
Baccino, T., 219, *225*
Bach, J., 168, *175*
Bachen, C. M., 767, *774*
Bachetti, E., 1056, *1058*
Bachmann-Medick, D., 1064, 1072, *1077*
Back, M., 170, *172*, *196*, 262, *264*
Bäckman, L., 814, 815, *827*
Bacon, D. J., 626, *637*
Badcock, D. R., 623, *637*
Baddeley, A. D., 52, 52*f* , 53, *59*, 69, 70, *74*, 250, *263*, 395, *410*, 420, 422, *428*, 575, 576, *597*
Bader, A., 1111, *1124*
Badham, R., 1077, *1079*
Badiqué, E., 625, 633, *635*
Badler, N. I., 626, *635*
Badman, A. L., 840, *85*1, 862, *869*
Badre, A., 1155, *1168*
Baecker, R. M., 7, *22*, 436, *457*, 576, 583, *597*, 643, *657*, 1077, *1081*, 1152, *1168*
Baeg, S. C., 427, *429*
Baehring, H., 862, *868*
Baer, M., 262, *264*
Bagozzi, R. P., 15, *22*, 673, *676*
Bagwell, C., 254, *263*
Bahan, B., 887, *893*
Bahl, P., 172, *172*
Bahrick, H. P., 53, *59*
Baier, A., 1254, *1261*
Bailey, B. P., 439, *456*
Bailey, J. P., 652, *657*
Bailey, R. W., 1141, *1147*
Bain, K., 846, *849*, 891, 891*f*, *892*
Bainbridge, L., 576, 582, *597*
Baines, A., 723, *740*
Bains, M., 706, *716*
Bair, J., 10, *23*
Baird, F., 192, *196*
Baker, B., 840, *849*, 1257, *1265*
Baker, K. S., 1068, *1079*
Baker, R., 1156, *1168*
Balachandran, A., 549, *555*
Balakrishnan, H., 180, *198*
Balakrishnan, J. D., 45, 48, *59*
Balakrishnan, R., 163, 167, 168, 169, 170, 171, *172*, *173*, *174*, *176*, 182, 191, 194, *198*, *199*
Balandin, S., 825, *827*
Balasubramanian, V., 560, *571*
Balazs, B., 224, *226*
Balci, O., 139, *144*
Baldwin, L. E., 138, *146*
Bale, S., 769, *774*
Balfanz D., 648, 650, *655*
Balfe, E., 442, *457*
Ball, E., 642, *655*
Ball, G., 83, 86, *89*
Ball, L. J., 1063, *1081*
Ballagas, R., 270, *294*
Ballas, J. A., 106, *111*, *112*, 624, *636*, 1203, *1207*
Ballentine, B., 389, *390*
Balme, L., 1118, *1124*
Balogh, J., 382, *390*
Bälter, O., 435*n*, *455*
Baltes, P. B., 780, *790*
Balthazard, P. A., 15, *24*
Bandeen-Roche, K., 858, *869*
Bandura, A., 122, *130*, 137, 140, *144*, *145*
Banerjee, S., 833, *849*
Bangalore, S., 415, *428*
Banker, R. D., 6, *22*
Banks, S. B., 591, *599*
Bannon, L. J., 10, *22*, 736, *737*, *739*, 966, *987*, 1250, *1261*
Banse, R., 86, *89*
Bao, L., 180, *196*
Bar-Yam, Y., 490, *505*, 577, *597*
Bar-Yossef, Z., 564, *571*
Barab, S. A., 617, *618*
Barclay, J. R., 67, *74*, 1067, *1080*
Bardy, B. G., 632, *638*
Bareket, M., 633, *636*
Barenfeld, M., 498, *507*
Barfield, W., 702, *716*
Bargh, J. A., 83, *92*
Barker, D. T., 384, *391*
Barker, P. G., 250, *263*
Barker, R. T., 1136, *1147*
Barkhuus, L., 1255, *1261*
Bartlett, F. C., 496, *505*
Barlow, T., 630, *637*
Barnard, L., 310, *311*, 854, 861, *867*, *868*
Barnard, M., 428, *430*
Barnard, P. B., 14, *22*, 99, *113*, 285, *292*, 395, 403, *410*, *411*, 1057, 1060, 1130, *1149*
Barnes, P., 768, *774*
Barnes, V., 217, *225*
Baron, S., 1195, *1207*
Barr, F. E., 668, *677*
Barralon, N., 1118, *1124*
Barreto, A. B., 841, *849*, 854, *868*
Barrett, L. F., 79, 87, *89*
Barrett, R., 568, *572*
Barrick, M. R., 127, *130*
Barry, M., 669, *677*
Barthelmess, P., 428, *428*

Bartle, R., 683, 684*t*, *699*
Bartlett, F. C., 722, *737*
Bartlett, K. M., 628, *637*
Bartlett, M. S., 451, *455*
Bartolome, D. S., 1227, *1239*
Bashashati, A., 842, *850*
Basil, M. D., 81, *91*
Basney, J., 650, *655*
Bass, L., 498, *505*, 1119, 1121, *1124*
Bass, M., 204, *225*
Bass, R., 846, *849*
Bass, S. A., 782, *790*
Basso, A., 1254, *1263*
Basson, S., 846, 891, *892*
Bastien, J. M. C., 1176, 1180, *1188*, *1189*
Batarbee, K., 1070, *1080*
Batchelder, W. H., 972, *987*
Batchelor, D., 365, *377*
Bates, A. W., 605, *618*
Bates, D. W., 667, 668, *676*, *677*
Bates, J., 83, 87, *89*
Bates, R., 843, *849*
Bathiche, S., 166, *174*
Batongbacal, A., 1123, *1124*
Battarbee, K., 977, *987*
Battestini, A., 193, *196*
Battista, G. D., 531, *542*
Battiste, V., 735, *739*
Battle, J., 764, *773*
Bauchner, H., 666, *676*
Baudel, T., 163, 169, 170, *172*, *175*, 182, *196*, 1032, *1039*
Baudisch, P., 165, 167, 170, *172*, *173*, *174*
Baudisch, R., 168, 171, *172*
Bauer, B., 298, *311*
Bauer, D., 220, *225*
Bauhs, J. A., 88, *91*
Baumann, K., 202, *225*
Baumeister, L. K., 1205, *1206*
Baumer, D., 642, *655*
Baumgartner, V. J., 375, 376, *378*
Baus, J., 440, 451, *456*, *457*
Baxter B., 932, 934, 935, 936, 937, 940, *946*
Baxter, G. D., 110, *112*
Baxter, L., 181, *196*
Bay, S., 224, *225*, *227*
Bayat, M., 224, *225*
Bayerl, J., 29, *39*
Bayir, H, 667, *676*
Bayles, M. D., 1257, *1261*
Bayley, O., 277, *293*
Baylis, G. C., 33, *41*
Beach, L. R., 574, 576, 583, *597*
Beale, R., 269, *292*, 854, *867*
Beard, M., 472, *486*, 1254, *1261*
Beardsley, P., 171, *175*, 474, *485*
Beasley, J. W., 667, *677*
Beattie, W., 817, *828*
Beatty, R., 583, *601*
Beauchamp, T., 1255, *1262*
Beaudouin-Lafon, M., 51, *59*, 166, 170, *172*, *174*, 182, *196*, 259, *263*, 1020, 1022, 1023, 1024, 1031, *1038*, 1112, *1125*
Becerra-Fernandez, I., 653, *658*
Beck, A. M., 1255, *1261*
Beck, E. E., 1062, *1077*
Beck, K., 912, *915*, 962, *963*, 1020, *1038*, 1042, 1055, *1057*
Becker, M., 1255, *1266*
Becker, L. C., 1252, *1261*
Becker, M. H., 138, *146*
Becker, R. A., 531, *542*
Becker, S. G., 58, *62*, 192, *198*, 574, *602*, 865, *867*
Becker, T., 414, *431*
Beckles, B., 650, *655*
Beckmann, C., 1037, *1038*
Bederson, B. B., 538, *542*, 644, *656*, 795, 797, *807*, *809*, 1031, 1032, 1033, *1038*, 1245, *1261*
Beedle, M., 912, *915*
Beeson, I., 1065, 1069, 1070, *1077*
Beevis, D., 1195, 1200, 1202, 1205, *1206*, *1207*
Begault, D., 624, *635*
Begeman, M., 15, *22*
Begole, J., 438*f*, *546*, 547, *557*, 642, *655*
Beidernikl, G., 611, *618*
Beigl, M., 182, *198*, 277, *293*
Beirmann, A. W., 823, *827*
Bejczy, A. K., 723, *737*
Bekey, G. A., 151, *157*
Bekker, H. L., 668, 669, *676*
Bekker, M. M., 800, 801, *808*, *809*, 1181, *1189*
Bekkering, H., 35, 38, *39*
Bell, B., 1175, *1189*
Bell, G., 946, *947*
Bell, M., 262, *263*, 285, *292*
Bellamy, R. K. E., 1054, *1058*
Bellon, O. R., 865, *869*
Bellotti, V., 170, *172*, 186, 191, *196*, *199*, 276, 277, *292*, 548, 551, *555*, *558*, 643, 647, *655*, 909, *915*, 986*n*, *987*, 1253, *1261*
Bellugi, U., 886, *893*
Belson, W., 1153, *1168*
Ben-Joseph, E., 475, *486*
Benard, M., 875, *881*
Benbasat, I., 15, 18, *22*, 138, *146*, 583, *600*, 669, *678*
Benford, S., 277, 285, *292*, *293*, 554, *555*, *556*
Bengio, S., 428, *428*, *430*
Bengler, K., 710, *718*
Beniger, J. R., 138, *145*
Benigno, D., 837, 849, *850*
Benko, H., 167, *173*
Bennett, C. J., 1247, *1261*
Bennett, J., 782, *791*, 934, *947*
Bennett, J. B., 1046, *1058*
Bennett, J. L., 9, *22*
Bennett, K. B., 121, *130*, 576, *597*, *601*
Bennion, F. A. R., 1257, *1261*
Benoit, C., 414, 415, 417, 420, 423, 427, *428*, *431*
Bentley, R., 642, *656*, 908, *915*, 966, 982, *986*
Bentley, T. D., 667, *677*
Benyon, D., 575, *600*, 742, *759*, 1177, *1188*
Berard, F., 184, *196*
Berbaum, K. S., 628, *637*
Berdahl, J. L., 547, *555*
Berdichevsky, D., 134, 144, *145*
Beregovaia, E., 1172, *1189*
Bereiter, C., 805, *808*
Berendt, B., 643, *658*
Berg, R., 485, *487*
Berg, S., 908, *915*
Bergel, M., 1137, *1149*
Berger, N., 1030, *1038*

Berglund, B., 250, *263*
Bergman, M. K., 561, *571*
Bergmann, M., 644, *657*
Bergqvist, U., 218, *225*, 314, *324*
Beringer, D. B., 726, *737*
Berkovitz, J., 795, *807*
Berkow, R., 832, 835, *849*
Berlage, T., 288, *292*
Berleur, J., 1257, *1261*
Berman, J. S., 87, *89*
Berman, M. L., 28, *39*, 163, *173*
Bernard, H. R., 969, 970, 971, *986*
Bernard, J., 419, *431*
Bernard, M., 796, *807*
Bernard, P. B., 314, *324*
Bernardin, K., 428, *429*
Bernaschi, M., 647, *655*
Berndtsson, J., 733, *738*
Berner, E., 595, *597*
Berners-Lee, T., 547, *555*, 564, *571*
Bernsen, N. O., 394, 400, 401, *410*
Bernstein, L., 415, *428*
Bernston, G. G., 85, *89*
Berry, G., 417, *431*
Berry, J. D., 338, *353*
Bers, J., 415, 426, *428*, 863, *869*
Bers, M. U., 1256, *1261*
Berscheid, E., 767, *774*
Bertelsen, O. W., 1048, *1057*, 1075, *1077*
Bertelson, P., 51, *59*
Berti, S., 1111, *1124*
Bertin, J., 403, *410*, 513, 514*t*, 517, 517*t*, 522, 523, 526, 531, *542*
Bertram, D., 581, *601*
Berzowska, J., 475, 481, *486*
Beshers, C., 518, 534, *542*
Bessiere, K., 563, *572*
Best, P., 218, *225*
Bettman, J. R., 673, *677*
Beu, A., 1146, *1148*
Beukelman, D. R., 816, *827*
Bevan, N., 859, *870*, 1075, *1078*, 1141, *1148*
Beveridge, M., 876, *881*
Bewley, W. L., 17, *22*
Beyer, H., 729, *738*, 907, 908, 909, 909*t*, *915*, *916*, 922, *926*, 934, *946*, *947*, 950, *963*, 978, *986*, 1047, *1057*, 1062, 1063, 1068, 1073, *1077*
Beymer, D., 181, 187, *196*
Bezerianos, A., 171, *173*
Bézivin, J., 1115, 1116, 1121, *1124*, *1125*
Bhabha, H. K., 1064, 1065, 1073, 1074, *1077*
Bhalodia, J., 547, *557*
Bharat, K., 814, *828*
Bhargav, A., 54, *61*
Bhattacherjee, A., 653, *655*
Bhavnani, S. K., 1202, 1205, *1206*
Bias, R. G., 1177, *1188*, 1268, 1270, 1276, *1277*
Bibby, P. A., 66, 67, *74*
Bichsel, M., 475, *487*
Biderman, A., 475, *486*, *487*
Bieber, M., 560, 561, *571*
Bieger, G. R., 403, *410*
Bier, E., 169, *173*, 1033, *1038*
Bierria, T., 873, 875, 876, 877, 878, 879, *881*
Biers, D. W., 1135, 1136, 1137, 1139, *1147*, *1148*
Biever, W. J., 715, *716*
Bigby, J. A., 674, *677*
Biggs, S. J., 624, *635*
Bihan, D., 1234, *1238*
Bikson, T. K., 547, *555*
Billinghurst, M., 182, *197*
Billings, C. E., 574, 582, 583, *597*, *601*, 727, 728, 729, 730, 734, *738*
Billsus, D., 441, 442, 449, *455*
Bilmes, J. A., 181, *196*
Binder, T., 1067, 1071, 1073, *1077*, *1079*
Biocca, F. A., 84, *91*, 224, *225*, 630, *637*
Birbaumer, N., 841, *849*
Birch, G. E., 838, 841, 842, *849*, *850*, *851*
Birget, J., 649, *658*
Birnbaum, L., 442*n*, *455*
Birren, J. E., 778, *790*
Birru, M., 873, 875, 876, 877, 878, 879, *881*
Bisantz, A. M., 575, *597*
Bischof, W. F., 633, *636*
Bishop, C. M., 184, 185, *196*
Bishop, D., 80, *89*
Bishop, G., 183, *198*
Bittner, A. C., 1196, *1207*
Bittner, K., 574, *597*
Bizantz, A., 577, *597*
Bizzi, E., 634, *635*, *636*
Bjerknes, G., 15, *22*, 1062, 1063, *1077*, 1250, *1261*
Björgvinsson, E., 1071, *1077*
Bjork, R. A., 70, 71, 72, *74*, *75*, 578, *597*
Björk, S., 224, *225*, 285, *292*
Bjorset, H., 218, *225*
Black, A., 137, *146*
Black, R. W., 614, *618*
Blackmon, M. H., 103, *111*, *112*
Blackwell, A., 2, *22*, 270, *294*, 649, *658*
Blackwood, W., 310, *311*
Blanco, M., 715, *716*, 878, *883*
Blandford, A., 502, *505*
Blanford, A., 645, *655*
Blank, H., 578, *597*
Blankenbach, K., 211, *225*
Blascovich, J., 785, 786, *790*
Blattner, M., 248, 250, 252, 256, 257, 257*f*, 258, *263*, *264*, 417, *430*, 863, *867*, 1102, *1105*
Blauert, J., 249, 252, 255, *263*
Blendon, 662, *678*
Blenkhorn, P., 843, *850*
Bless, H., 81, *91*, 116, *130*
Blin, O., 48, *59*
Bliss, M. E., 781, *790*
Blocher, A., 414, *431*
Blom, J., 1256, *1265*
Blomberg, J. L., 966, 971, 973, 980, 982, *986*, *987*, *988*, 1063, 1066, 1068, *1077*, *1079*, *1080*
Blomkvist, S., 609, *618*
Blomquist, Å., 1015, *1015*
Bloom, T., 168, *176*
Bloor, C., 837, 842, 849, *850*
Blount, S., 578, *597*
Blum, R., 665, *677*
Blumberg, B., 170, *175*, 180, *197*
Blumler, J. G., 88, *90*
Blundel, B. G., 204, *225*
Bly, S. A., 249, *263*, 277, *293*, 552, 553, *555*, *557*
Blythe, M. A., 17, *22*, 194, *196*

Boal, A., 1071, *1077*
Bobick, A., 192, *196*
Bobrow, D. G., 124, *131*, 276, *294*
Bochman, A., 596, *597*
Bødker, K., 575, *597*, 982, *987*, 1063, 1067, 1068, *1078*, *1079*, 1250, *1264*
Bødker, S., 21, *22*, 1024, 1025, *1038*, 1047, 1048, *1057*, 1062, 1065, 1071, 1072, 1075, *1077*, *1078*, 1250, *1261*
Boehm, B. W., 502, *505*, 1021, *1038*
Boehm-Davis, D. A., 1203, *1207*
Boehner, K., 1251, *1265*
Bogart, E. H., 1227, *1239*
Bogdan, C., 277, *294*
Bohman, P., 874, *881*
Boies, S. J., 497, *506*
Boissiere, P., 845, *849*
Bojko, A., 1139, 1140, *1147*
Boldini, A., 46, *59*
Bolle, R. M., 414, *431*
Bolles, R. C., 184, *196*
Bolt, R. A., 170, *173*, 414, 422, 425, 426, *429*
Bolter, J. D., 1037, *1039*
Boltman, A., 1065, 1069, 1070, *1078*
Bolton, R., 1070, *1077*
Bonacker, M., 218, *226*
Bond, M. R., 832, *851*
Boneva, B., 549, *556*
Bonin-Guillaume, S., 48, *59*
Bonner, M. C., 1233, *1238*
Bontcheva, K., 444, *455*
Booch, G., 1055, *1058*
Booher, H. R., 401, *410*
Booker, J., 1056, *1058*
Boone, K., 171, *174*
Boorstin, J., 681, 684*t*, 685, *699*
Booske, B. C., 671, 672, *676*, *678*
Booth, B., 71, *75*
Booth, K. S., 436, *457*, 522, *542*, 797, *807*
Booth, L., 817, 822, 828
Boothby, R., 220, *226*
Borchers, A., 649, *656*
Boren, M., 1133, 1140, *1147*
Borgatti, S., 610, *618*
Borgman, C. L., 1248, *1261*
Borisoff, J. F., 842, *850*
Boritz, G. A., 784, 785, *790*
Borman, L., 11, *22*
Borning, A., 1250, 1251, 1255, *1261*, *1262*, *1266*
Borovoy, R., 485, *487*
Borriello, G., 171, *176*, *197*
Boruff, B., 52, *61*
Bos, N. D., 553, *555*, *557*, *558*, 1254, *1261*, *1266*
Boschmann, M., 219, *225*
Bosman, D., 209, 212, 213, *225*
Bosman, E. A., 784, 786, *790*, 814, *827*
Bosquet, D., 69, *76*
Bost, R., 1200, *1206*
Bothell, D., 96, 106, *111*, 575, *597*
Bouas, K. S., 547, *555*
Boucher, A., 1036, *1038*
Boucher, L., 38, *40*
Bouchette, D., 1131, *1149*
Bouillon, L., 1119, 1123, 1124, *1124*, *1125*
Boulanger, P., 633, *636*
Bourg, W. J., 138, *146*
Bourne, L. E., Jr., 53, *59*
Bovair, S., 66, *75*, 98, 99, 100, *111*, *112*, 502, *505*, 1205, *1207*
Bowen, C. D., 1178, *1188*
Bower, G. H., 44, *59*, 79, 81, *89*
Bowers, C., 574, 583, *597*, *601*, 624, 626, *636*, *638*
Bowers, J., 277, *292*, *294*
Bowker, G. C., 1068*n*, *1079*
Bowman, D., 162, *173*
Bowman, T., 841, *850*
Bowman, W. J., 17, *22*
Boxer, S., 365, *377*
Boyd, J., 182, *196*
Boyd, R. A., 1246, *1261*
Boydell, O., 442, *457*
Boyer, D. G., 643, *656*
Boyer, J., 878, *883*
Boyes-Braem, P., 1043, *1059*
Boyle, D. L., 667, 668, *676*, *678*
Boyle, M., 1253, *1261*
Bozinovska, L., 83, *89*
Bozinovski, S., 83, *89*
Bozorgzadeh, Z., 838, 841, *849*, *850*, *851*
Braa, K., 1066, *1077*
Brachman, R., 578, 595, *597*
Bradburn, N. M., 754, *758*, *759*
Bradfield, A., 578, 595, *597*
Bradley, S. K., 633, *636*
Bradner, E., 549, *556*, 1254, *1261*
Brady, S., 887, *892*
Braeuninger, U., 321, *325*
Braffort, A., 890, *892*
Brailsford, T. J., 568, 569, 570, *571*, *572*
Brailsford. D., 564, *571*
Brain, D., 366, *377*
Brainbridge, 728, *737*
Branaghan, R. J., 1074, *1080*, 1131, *1147*
Brandeis, L. D., 642, *658*
Brandt, E., 1067, 1069, 1071, 1072, 1073, *1077*, *1079*
Bransford, J. D., 53, *60*, 67, *74*, 496, *505*
Brashear, H., 182, *196*, 890, *892*, *893*
Brashers-Krugg, T., 634, *635*
Bratteteig, T., 1062, *1077*, 1250, *1261*
Braun, M., 625, *635*
Brave, S., 284, *294*, 473, 474, 476, *486*, 480, 682, *699*, 1253, *1264*
Brawner, P., 862, *868*
Bray, T., 541, *542*
Breaux, R., 625, *637*
Breckler, S. T., 87, *89*
Bredeweg, B., 445*n*, *457*
Breese, J., 83, 86, *89*, 172, *174*, 191, *197*, 453, *456*
Bregler, C., 417, 426, *429*
Brehm, B., 138, *145*
Brehm, S., 576, 579, *597*
Brehner, B., 574, *600*
Breiman, L., 186, *196*
Breining, R., 625, *635*
Brennan, C. W., 33, *40*
Brennan, P. F., 782, *791*
Brenner, M. H., 858, *867*, *868*
Breuker, J., 445*n*, *457*
Brewer, E., 646, *658*
Brewer, J., *870*, 888, *892*
Brewster, S. A., 168, *175*, 257, 260*f*, 261, *263*, *264*, 286, *292*, 839, *851*, 854, 859, 862, *867*, *869*, *870*

Bricker, P. D., 548, *556*
Brickman, B., 575, *597*
Briefs, U., 1063, *1077*
Brienza, D. M., 835, 838, 843, 847, *851*
Briggs, C. A., 323, *325*, 972, *986*
Briggs, J., 1135, *1147*
Briggs, P., 442, 457, 652, 653, *655*, *657*, *658*
Briggs, R. O., 15, *24*, 1022, *1038*
Brin, S., 567, *571*
Brinck, T., 276, *293*
Bringhurst, R., 335, *353*
Brinkley, J. F., 667, *676*
Britt, M. A., 69, *76*, 877, *881*
Britton, B. K., 69, *76*
Britton, C., 846, *850*
Britton, E., 167, *173*
Britz, J. J., 1252, *1264*
Broadbent, D. E., 28, 44, 54, *39*, *59*
Broder, A. Z., 564, *571*
Brodie, C., 640, 642, 643, 645, 652, 653, *655*, *657*
Brodie, J., 224, *225*, 819, *828*
Brodskiy, A., 649, *658*
Brooke, J., 1138, *1147*
Brooke, N. M., 417, *429*, *432*
Brooks, C., 595, *597*
Brooks, F. P., Jr., 167, *175*, 232, *244*, 1043, 1046, 1048, *1057*
Brooks, V. B., 29, *39*
Brosnan, M. J., 765, *774*
Brostoff, S., 647, *655*
Brouwer, W. H., 858, *867*
Browall, C., 183, *197*
Brown J., 770, *773*
Brown, A. B., 492, 493, 495, 503, *505*
Brown, B. R., *292*, *378*
Brown, C. M., 554, *555*, 833, 874, 879, *881*, 1223, 1238
Brown, D., 874, *881*
Brown, G. D. A., 492, *505*
Brown, J., 52, *59*
Brown, J. M., 121, *130*
Brown, J. S., 1256, *1266*
Brown, L., 642, *658*
Brown, R., 308, *311*, 833, *851*
Brown, S., 56, *59*, 683, *700*
Brown, W., 415, *429*
Browne, P., 565, *571*
Bruce, R., 550, *555*
Bruckman, A. S., 499, *505*, 552, *555*, 804, 805, 806, *807*, *808*, 1259, *1263*
Bruckmann, D., 1067, *1080*
Brudnicki, D. J., 734, *738*
Bruin, J. D., 521, *542*
Brumitt, B. L., 170, *174*, 188, 193, *196*, *197*
Brun-Cotton, F., 971, *986*
Brunnstein, K., 1257, *1261*
Brusilovsky, P., 441*n*, *455*, 568, 569, *571*
Bruynzeel, M., 634, *636*
Bryan, L. A., 726, *738*
Bryant, S. L., 552, *555*
Brzezinski, J., 83, *89*
Bub, U., 426, *432*
Buchanan, G., 171, *174*
Buchanan, M., 52, *59*
Buckman, R., 902, *915*
Buckwalter, G. J., 634, *637*
Buderi, R., 1218, *1218*
Budicks, D., 859, *869*
Budin J.-P., 214, 215, *225*
Budzik, J., 442*n*, *455*
Buff, W., 626, *638*
Buffett, S., 644, *655*
Buford, B., 652, *655*
Bühler, C., 864, *867*
Bui, X. T., 1057, *1058*
Bull, R., 495, *506*
Buller, D. B., 141, *145*
Bulliet, R., 1243, *1261*
Bullinger, A. H., 138, 139, *145*
Bullinger, H.-J., 625, 626, *635*
Bulthoff, H., 422, 428, *429*
Bunch, C., 1062, *1077*
Bungert, C., 623, *635*
Bunt, A., 436, 444, *455*
Buourgeois, F., 166, *174*
Burbeck, S., 472, *486*
Burbules, N. C., 142, *145*
Burchard, P., 539, *543*
Burdea, G. C., 167, 168, *173*, 230, 233, *244*, 624, 626, 635, *635*, *637*
Burdick, M., 576, *601*
Burge, E. L., 609, *618*
Burgoon, J. K., 141, *145*
Burke, R. D., 442, *455*
Burke, S. J., 643, *657*
Burmester, M., 684*t*, *699*, 1146, *1148*
Burnett, G. E., 563, *571*, 711*t*, *718*
Burns, C., 577, 580, *597*
Burns, J., 440, *458*
Burns, M., 629, *638*
Burr, B. J., 28, *39*, 51, *59*, 163, 166, *173*
Burroughs, M., 723, *738*
Bursill, A. E., 118, *130*
Busch, C., 117, *131*
Bush, V., 4, *22*, 546, *555*
Butler, C. W., 502, *506*
Butler, K., 1212, *1218*
Butler, M. B., 937, *946*
Butler, M. H., 1249, *1266*
Butler, R. A., 624, *635*
Butler, S., 1141, *1148*
Butterfield, E. C., 44, *60*
Button, G., 982, *986*
Butxon, W., 166, *174*
Buur, J., 1062, 1065, 1066, 1067, 1071, 1072, 1073, 1074, *1077*, *1078*, *1079*, *1080*
Buxton, B., 167, 169, 171, *173*, *175*
Buxton, E., 163, *173*
Buxton, W., 7, *22*, 162, 163, 164, 164*f* , 165, 167, 170, 171, *173*, *174*, *175*, *176*, 180, *196*, 248, 249, 250, 251, 252, *263*, 277, *292*, *294*, 475, 484, 485, *486*, 576, *597*, 643, *657*, 844, *851*, 1032, 1033, *1038*, *1039*, 1152, *1168*
Buyukkokten, O., 171, 173
Buzan, B., 1068 , *1078*
Buzan, T., 1068, *1078*
Bynum, T. W., 1247, *1261*
Byrne, B., 976, *988*
Byrne, M. D., 96, 99, 100, 106, 109, 110, *111*, 575, *597*, 804, *807*, 1205, *1206*
Byrne, N., 575, *597*
Byun, H. E., 449, *455*

C

Cacioppo, J. T., 83, 85, 86, *89*, *90*, 141, *146*
Cadiz, J. J., 172, *173*, 188, *196*, 549, *555*, 1253, *1263*
Caggiano, D., 582, *597*
Cahn, J. E., 87, *89*
Cail, F., 322, *325*
Cailliau, R., 561, 562, *571*
Cain, M. M., 674, *677*, 873, *881*
Caird, J. K., 120, 123, *130*, *131*, 581, *598*
Cairncross, F., 554, *555*
Cairns, A. Y., 819, 823, 824, 825, *828*
Çakir, A., 6, *22*, 207, *225*, 314, 321, *324*
Calabresi, P. A., 33, *40*
Calder, J., 427, *429*
Calder, P. R., 1033, *1039*
Caldwell, B., 583, *597*
Callahan, J., 634, *636*
Caloyannides, M. A., 650, *656*
Calvary, G., 1118, 1119, 1121, *1124*
Calvert, G., 420, 422, 428, *429*
Calvert, K., 806, *807*
Calvert, S. L., 630, *635*
Calvo, M. G., 125, 128, *130*
Cameron, G. T., 138, *146*
Cameron, M., 1067, 1068, *1078*
Camp, L. J., 547, *555*
Campbell, A., 273, *292*
Campbell, C., 168, *173*, 568, *572*
Campbell, D. T., 1179, *1188*
Campbell, E. A., 121, 130
Campbell, J. L., 711*t*, *716*
Campbell, L., 192, 196
Campbell, M., 879, *882*
Campbell, R. H., 590, *598*, 1179, 1180, *1188*
Campbell, R. L., 14, *23*, 1057, *1057*, 1245, *1261*
Campbell, S. E., 121, *131*
Campion, M. A., 1160, *1169*
Candry, P., 211, *225*
Canker, M., 1138, *1148*
Cannon, W. B., 85, *89*
Cannon-Bowers, J. A., 72, *75*, 121, *131*
Cano, C., 475, *486*
Cantani, M. B., 1135, *1139*
Caplan, D., 496, *506*
Caplan, M., 754, *758*
Carames, R., 765, *774*
Carasik, R. P., 1075, *1078*
Carayon, P. C., 314, 315, 323, *324*, *325*, *326*
Carcillo, J. A., 667, *676*
Card, S. K., 7, 12, 14, 16, *22*, *23*, *24*, 28, 29, *39*, 44, 47, 51, 58, *59*, 94, 95, 96, 97, 97*f*, 98, 99, 108, *111*, *112*, 162, 163, 166, 167, *173*, *174*, 286, *292*, 395, *410*, 490, *505*, 515, 517*t*, 518, 519, 519*f*, 520*t*, 521*t*, 522*f*, 524*t*, 534, 538, 540, 541, *542*, *543*, 566, 567, *571*, *572*, 575, 576, *597*, 714, *716*, 839, *850*, 1152, *1168*, 1195, 1198, 1204, 1205, *1207*, 1214, *1218*
Carenini, G., 444, *455*
Carey, K., 549, *556*
Carey, T., 742, *759*
Caridakis, G., 889, *893*
Carini, R. M., 1167, *1168*
Carley, K., 547, *555*
Carlin, A. S., 139, *145*
Carlson, R., 575, *597*
Carlsson, C, 668, *678*
Carmein, D., 232, *244*
Carmel, E., 554, *555*
Carmichael, A. R., 814, 826, *827*, *828*, 872, *882*
Carmien, S., 1067, 1077, *1078*
Carney, C., 711*t*, *716*
Carney, P., 139, *146*
Carnine, D., 139, *146*
Carpendale, S., 194, *197*
Carpenter, P. A., 99, *112*, 876, *882*, 1223, *1238*
Carpenter, R., 427, *429*
Carr, L., 565, *571*
Carr, R., 1066, *1080*
Carretta, T. R., 631, *635*, *638*
Carrillo, R., 1064, *1078*
Carrithers, C., 497, *505*, 1054, *1057*
Carro, R., 570, *571*
Carroll, J. M., 14, *23*, 64, 65, 71, *74*, 361, *377*, 492, 497, 498, *505*, 575, 583, *597*, 876, *881*, 978, 980, *986*, 993, 1008, *1015*, 1043, 1044, 1045, 1046, 1047, 1049, 1052, 1053, 1054, 1055, 1056, 1057, *1057*, *1058*, *1059*, 1063, 1068, 1071, *1078*, *1079*, *1080*, 1116, *1125*
Carroll, K., 842, *850*
Carson, D., 628, *635*
Carson, R. G., 29, *39*
Carsten, O. M. J., 708, *718*
Carswell, C. M., 673, *676*, 724, *740*, 1144, *1149*
Cart, M., 288, *294*
Carter, K., 1066, *1079*, *1080*
Carter, M., 550, *557*
Carter, S., 1070*n*, 1078
Cartwright, D., 583, *597*
Caruana, R., 436, *457*
Caruso, D., 142, *145*
Caruso, M. J., 180, *196*
Carver, C. S., 116, 119*f*, 121, *130*, *132*
Cary, M., 575, *597*
Casali, S. P., 788, *790*, 835, 837, 847, 848, *850*
Case, J., 675, *678*
Casey, S., 149, *157*
Cash, D., 1253, *1261*
Cashen, M., 873, *881*
Casner, S., 517, *542*
Cassard, S. D., 858, *869*
Cassell, J., 78, 86, 87, *89*, 415, 417, *429*, *430*, 743, *758*
Cassidy, H., 220, *226*
Castel, A. D., 32, *39*
Castellani, S., 501, *506*
Castellano, J. A., 202, 208, 209, 211, 214, 215, *225*
Castiello, U., 37, *39*
Castillo, E. R., 633, *636*
Castronova, E., 608, *618*
Catani, M. B., 55, *61*, 1135, 1144, *1147*
Cathiard, M.-A., 423, *428*
Cattell, R. B., 814, *827*
Caulderwood, R., 577, *599*
Cavazza, M., 625, *635*
Cavender, A., 844, *850*
Cavonius, C. R., 220, *225*
Cawley, T., 565, *571*
Cawsey, A., 444, 451, *455*
Ceaparu, I., 563, *572*
Ceci, R., 139, *145*
Cerella, J., 788, *790*
Ceverst, K., 192, *196*

Chadwick-Dias, A., 55, *61*
Chai, J., 385, *390*
Chaika, M., 742, *758*
Chaiken, S., 653, *656*
Chakrabarti, P., 580, *598*
Challis, B. H., 53, *59*
Chalmers, M., 285, *292*
Chan, P., 165, *173*
Chan, T., 553, *556*
Chandler, J., 1076, *1078*
Chandler, P., 69, *74*
Chandra, C., 664, *677*
Chandrasekaran, B., 581, *597*
Chang, A., 474, 475, 476, *486*
Chang, S. K., 579, *602*, 840, *849*, 862, *866*, *867*
Chapanis, A., 29, *39*, 148, 150, *157*, 724, *738*, 1026, *1038*
Chapman, P., 706, *716*
Chapman, R., 574, 583, 601
Chapman, V., 19, *23*
Chappel, K. K., 767, *774*
Chapuis, O., 1033, *1038*
Charles, L., 273, 275, 276, *277*, 878, 879, *881*
Charness, N., 581, *597*, 778, 784, 785, 786, 787, 788, *790*, *791*, *792*, 814, *827*
Charron, C., 980, *987*
Chase, G., 565, *571*
Chase, J. D., 788, *790*, 837, *850*
Chassin, M. R., 662, *676*
Chattratichart, J., 224, *225*
Chatty, S., 259, *264*
Chavan, A. L., 360, *377*, 939, *946*
Chawathe, Y., 179, *197*
Cheal, M. L., 31, *39*
Checkland, P. B., 1046, *1058*, 1062, 1063, *1078*
Chellappa, R., 185, *199*
Chen, B., 377, 380
Chen, C., 1054, *1058*
Chen, D., 288, *294*, 702, *719*
Chen, E., 377, 378
Chen, F., 423, *429*
Chen, J. L., 569, *572*, 628, *635*
Chen, R., 876, 883
Cheng, C. H., 667, *676*
Cheng, D., 181, 198
Cheng, K., 563, *572*
Cheng, P. C. H., 109, *112*
Cheng, T., 874, *882*
Chernoff, H., 495, *505*
Cherny, L., 1052, *1058*
Cherri, C., 708, 717
Cherry, E. C., 30, *39*, 54, *59*
Chesbrough, H., 1212, *1218*
Cheverst, K., 449, *455*
Chewar, C. M., 1056, *1058*, *1059*
Cheyer, A. J., 415, *429*, 427, *430*
Chi, E. H., 109, *111*, 519, 540, *542*
Chiang, Y. T., 606, *619*
Chickering, D. M., 187, *197*
Chidambaram, N., 280, *292*
Chignell, M. H., 120, 121, 122, 129, *131*, 861, *867*, *870*
Childress, W., 137, 138, *146*
Chimani, M., 437, *457*
Chin, G., 1044, 1045, 1049, 1052, *1057*, *1058*, 1074, *1078*
Chin, J. P., 87, *89*, 1138, *1147*
Chiou, G. F., 608, 619
Chipman, G., 799, *807*
Chipman, S., 575, *601*
Chirst, S. E., 31, *39*
Chisholm, M., 307, *311*
Chisnell, D., 1143, *1147*
Chiu, P., 165, *175*, *551*, 556
Cho, Y. S., 50, *59*
Choi, J. H., 604, *618*
Chong, R. S., 111, *111*
Choo, S., 475, 486
Choong Y., 361, *377*
Chorost, M., 886, *892*
Choudhury, T., 181, *196*, *197*, 417, *429*
Chow, C., 103, 111
Chowning, J., 253, *263*
Christakos, C. K., 309, *311*
Christensen, J. R., 232, *244*, 833, *851*, 1255, *1263*
Christerson, M., 993, *1016*
Christie, B., 395, *410*, 549, *557*
Christie, M. A., 803, *807*
Christopher, J. C., 1245, *1261*
Chrysler, S. T., 67, *75*
Chu, J. Y. M., 854, *868*
Chua, R., 29, *39*
Chua, P. T., 634, *635*
Chua, R., 28, 29, 33, 35, 36, *39*, *40*, *41*
Chuah, M. C., 536, *542*
Chubb, G. P., 1194, *1207*
Chudy, A., 735, *739*
Chun, B., 194, *197*
Chung, L., 1055, *1059*
Church, D., 628, *636*
Churchill, E., 78, *89*, 417, *429*
Chute, D. L., 781, *790*
Cianchette, C., 55, *61*, 1137, *1149*
Ciborra, C., 1063, *1077*
Ciesielski, K. T., 1234, *1238*
Ciuffreda, K., 219, *226*
Clancey, W. J., 580, 583, *597*, *598*
Clark, H. H., 72, *74*, 382, *390*, 404, *410*, 1044, *1058*
Clark, J. M., 69, *74*
Clark, L. A., 87, *92*
Clark, R. S. B., 667, *676*
Clark, S., 845, *852*
Clark, W., 4, *23*
Clarke, A., 218, *227*
Clarke, D., 274, *292*
Clarke, K., 583, *597*
Clarkson, B., 417, *429*, 451, *457*
Clarkson, E., 168, *173*, 303, *311*
Clarkson, J., 837, 839, 840, 847, *850*
Clauer, R., 552, 557
Clausen, C., 357, *377*
Clauss, S., 365, *657*
Clawson, J., 168, *173*, 303, 304, *311*
Clegg, B. A., 44, *62*
Clement, A., 1063, 1067, *1078*, *1080*, *1081*
Clements, D. H., 804, *807*
Clements, P., 1121, *1124*
Cleveland, W. S., 517, *542*
Clifford, C. W. G., 491, *507*
Clifford, J., 968, *986*
Clinchy, B. M., 612, *618*

Cline, R. J. W., 781, *790*
Clore, G. C., 80, 81, 83, *89*, *91*, 682, *699*
Clore, G. L., 405, *411*
Clow, J., 419, *429*
Cobb, S., 1065, 1078
Coble, J. M., 924, *926*
Coble, J. R., 625, *637*
Cochran, W., 1166, *1168*
Cockayne, W., 232, *244*
Cockburn, A., 65, *74*, 574, *597*, 912, *915*, 962, *963*
Cockton, G., 837, 842, 849, *850*, 1110, *1124*, 1131, *1147*, 1172, 1173, 1174, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, *1188*, *1189*, *1190*
Codella, C., 426, *429*
Cody, M. J., 780, 785, *790*
Coeckelbergh, T. R. M., 858, *867*
Coetzee, F. M., 564, *572*
Coffetti, P., 650, *657*
Cohen, A. L., 667, *677*, 1253, *1261*
Cohen, B. G. F., 314, *326*
Cohen, E., 723, *738*
Cohen, G., 821, *827*
Cohen, J., 474, *486*
Cohen, K., 969, *986*
Cohen, M. M., 262, *263*, 382, *390*, *429*, *430*, 624, *636*, 667, *676*
Cohen, P. R., 169, *173*, 414, 415, 417, 419, 420, 424, 426, 427, 428, *429*, *430*, *431*, *432*, 861, 863, *869*
Cohen, W. J., 314, 315, *324*, *326*
Cohen, Y., 31, 32, 33, *40*, 55, *60*
Cohn, J. F., 85, *92*
Cohn, J. V., 633, 634, 636, *637*
Coiffet, P., 635, *635*
Cole, C. A., 674, *677*
Cole, D. E., 702, *718*
Cole, E. L., 724, *738*, 879, *881*
Cole, M., 802, *808*, 1243, *1261*
Cole, S. T., 1167, *1168*
Colella, V., 485, *487*
Colenbrander, A., 855, *867*
Coles, M. G. H., 47, *61*, 116, *131*
Collaros, P. A., 1022, *1038*
Colley, A. N., 764, 765, *774*
Collier, G., 86, *89*
Collins, A., 64, *74*, 83, *91*, 405, *411*
Collins, V., 877, *882*
Colonius, H., 48, *61*
Colton, T. C., 873, *882*
Comaroff, J., 969, *986*
Comber, C., 765, *774*
Compton, D. E., 625, *637*
Comstock, E., *934*, 947
Conati, C., 436, 444, *455*
Condon, A., 581, *600*
Condon, W. S., 415, *429*
Coninx, K., 1123, *1124*
Conklin, P. F., 1270, *1277*
Connell, I. W., 502, *505*, 1181, *1188*
Connell, J., 642, *658*
Connolly, T., 546, *556*, 574, 576, 583, *597*, 723, *738*
Conrad, F. G., 96, *111*, 1167, *1168*
Conrad, R., 723, *738*, 887, *892*
Considine, M., 385, *391*
Consolvo, S., 179, *197*, 1253, 1255, *1262*, *1263*
Constantine, L. L., 993, *1015*, 1048, *1058*, 1065, *1079*
Conti, G., 650, *656*
Contractor, N., 583, *599*
Conventry, L., 649, *656*
Converse, J. M., 1153, *1168*
Conversy, S., 259, *263*
Conway, F. T., 854, *868*
Conway, G. E., 117, *132*
Conway, M., 162, *174*
Cook, D., 610, *618*
Cook, M., 549, *555*
Cook, P., 15, *22*
Cook, T. D., 1179, *1188*
Cooke, N. J., 58, *60*, 88, *91*
Coombs, N., 862, *867*
Cooper, A., 609, *618*, 941, *946*, 956, *963*, 993, *1015*, 1054, *1058*
Cooper, G. E., 732, *738*
Cooper, J., 764, 765, 767, 768, 769, 770, 771, 773, *774*, *775*
Cooper, M., 1254, *1262*
Corazza, S., 888, *892*
Corbato, F., 4, *23*
Corbett, A. T., 96, *111*, 445, 446, *455*
Corcos, E., 822, *828*
Cordero, L., 668, *676*
Corley, M., 877, *882*
Cornelissen, F. W., 858, *867*
Cornsweet, D. M., 118, *130*
Cornwall, R., 1223, *1238*
Corral-Verduga, V., 970, *986*
Corritore, C., 652, 653, *656*
Cortes, A., 624, *636*
Coskun, E., 491, *506*
Cosley, D., 553, *555*
Cosmides, L., 79, *92*
Costa, P. T., 121, 127, *131*
Costantino, T., 1067, *1080*
Cothran, J., 261, *264*
Cottam, M., 912, *915*
Cotton, B., 734, *738*
Cotton, J. L., 1077, *1078*
Cottrell, G. W., 492, *507*
Coulouris, G., 287, *294*
Coulson, G., 283, *293*
Coulston, R., 395, *411*, 420, 421, 422, *431*, *432*
Coulter, K. J., 96, *112*
Couper, M. P., 754, *758*, 1167, *1168*
Courage, C., 932, 934, 935, 936, 937, 940, *946*
Courtire, A., 51, *59*
Courtney, J. F., 668, *678*
Coutaz, J., 1108, 1118, 1119, *1124*, *1125*
Coventry, L., 650, *656*
Covi, L. A., 551, *557*
Covington, M. V., 128, *130*
Cowie, R., 85, 86, *89*
Cox, S., 889, *892*
Coyle, E., 491, *507*
Coyle, M., 442, *457*
Coyne, K. P., 1139, 1143, *1147*, *1149*
Crabtree, A., 285, *292*, 982, *986*, 1066, 1068, *1078*
Craig, A., 841, *850*, *851*
Craig, J., 338, *353*
Craig, K. M., 547, *555*
Craik, F. I. M., 53, *59*
Craik, K. J. W., 66, *74*, 730, *738*
Crampton Smith, G., 474, *486*
Crandall, B., 575, 577, 579, 581, *597*
Crane, P. M., 734, *738*

Crane, W., 858, *867*
Cranor, L., 448, 449, 450, *455*, 642, 643, 644, 645, 647, *655*, *656*, *657*, 1244, 1245, 1251, 1252, *1261*, *1262*
Craven, J., 854, 864, 865, 866, *867*
Crawford, A., 549, 556
Crawford, C., 743, *758*
Crawford, E., 435, *456*, *457*
Crawford, G., 610, *618*
Crawford, S. D., 1167, *1168*
Crayne, S., 436, *456*
Crease, M. C., 259, *263*
Creelman, C. D., 45, *60*
Creemers, B., 1123, *1124*
Crisman, E., 843, 849, *852*
Crispien, K., 863, *867*
Cristea, A., 570, *571*
Cristianini, N., 186, *196*
Critchley, M., 821, *827*
Crivella, R., 634, *635*
Crocca, W. T., 1075, *1077*
Crocker, M. J., 323, *324*
Croft D. R., 875, *882*
Cromley, J., 877, *882*
Cronbach, L. J., 127, *130*, 1162, *1168*
Crosbie, R. E., 139, *145*
Croson, R., 578, *597*
Cross, N., 1042, 1043, *1058*
Cross, R., 972, *986*
Crossman, E. R. F. W., 29, *39*
Crowell, C., 139, *146*
Crowfoot, N. C., 1066, *1080*
Crowley, J. L., 184, *196*, 1108, *1124*
Crowston, K., 73, *75*
Crundall, D., 706, *716*
Cruz, I. F., 531, *542*, 1153, *1169*
Cruz-Neira, C., 623, *636*
Csikszentmihalyi, M., 122, 123, *130*, *132*, 684, 684*t*, 686, 687*f*, *699*, 747, *758*
Cullen M., C., 970, 974, *987*
Culnan, M., 640, *656*
Cummings, A., 548, 551, *555*
Cummings, J. N., 548, *556*
Cummings, M. L., 1251, *1262*
Cunningham, C., 862, *867*
Cunningham, E. T., Jr., 856, *868*
Cunningham, W., 1055, *1057*
Cuomo, D. L., 1178, *1188*
Curbow, B., 858, *867*, *868*
Curran, E., 183, *196*
Curry, R. E., 590, *602*, 713, *716*, 728, 729, *740*
Curson, I., 1141, *1148*
Curtis, B., 73, *75*
Cuschieri, A., 37, *39*
Cutler, L. D., 474, *486*
Cutrell, E., 168, *172*, *173*, *174*, *175*
Cypher, A., 804, *807*
Czaja, S. J., 778, 779, 780, 781, 782, 784, 785, 786, 787, 788, *790*, *791*, *792*
Czarnecki, K., 1117, *1125*
Czerwinski, M., 164*f*, *174*, 406, *411*, 878, *882*, 1253, *1266*

D

D'Alessandro, D. M., 875, *882*
D'Andrade, R. G., 970, *986*
Dâassi, O., 1121, *1124*
Dahlbäck, N., 390, *390*, 420, *429*
Dahlberg, T., 673, *676*
Dahley, A., 284, *294*, 473, 474, 475, 476, *486*
Dai, L., 844, *850*
Dainoff, M. J., 321, *325*
Dair, C., 342*f*, 353
Dalke, A., 415, *432*
Dallas, M., 53, *60*
Dalrymple, M., 414, 415, *429*
Dalwood, M., 565, *571*
Daly, B., 628, *635*
Daly, L., 634, *636*
Daly-Jones, O., 549, *555*, 1075, *1078*
Damasio, A. R., 79, *89*, 680, 681, 682, *699*
Dambrot, F. H., 764, *774*
Damodaran, L., 6, *23*
Dandavate, U., 1070, 1074, *1078*
Daneman, M., 876, *882*
Danielson, D. R., 142, 143, *145*, 652, *656*, 877, 878, *882*
Danielsson, N., 652, *657*
Danis, C. M., 384, *391*, 500, *507*, 862, *867*
Dannenberg, R. B., 248, *263*, 1035, *1039*
Danninger, M., 428, *429*
Danowski, J. A., 604, *618*, 785, *790*
Dardailler, D., 870, 888, *892*
Dark, V. J., 576, *599*
Darken, R. P., 232, *244*, 628, 633, *636*, *637*
Darley, J., 770, *775*
Darrach, B., 8, 9, *23*
Darrell, T., 170, *175*, 180, *197*, *198*, 414, 417, *430*, *431*
Darves C., 395, *411*
Darvishi, A., 862, *867*
Darwin, C., 79, *89*
Dascola, J., 33, *40*
Dasgupta, P., 580, *598*
Dashevsky, S. G., 724, *738*
Datz-Kauffold, B., 1135, *1147*
Daubman, K. A., 81, *90*
Daude, R., 217, *227*
Dave Pelz, 498, *506*
Dave, K., 552, *558*, 606, *619*
Davenport, A., 794, *807*
Davenport, D., 172, *176*
Davenport, W. W., 727, *738*
David, R. M., 833, *852*
David, S., 1251, *1265*
Davidson, A. I., 1076, *1078*
Davidson, R. J., 80, 85, *89*, *91*
Davies, D. R., 45, *60*, 786, *792*, 798, *882*
Davies, M., 394, *411*, 798, *808*
Davies, N., 192, *196*, 277, *292*
Davies, R. C., 622, *636*, 1077, *1078*
Davies, S., 1176, *1189*
Davies-Adetugbo, A., 1250, *1264*
Davis, D., 649, *656*
Davis, F. D., 6, 15, *22*, *23*, *676*, 901, *915*
Davis, G. B., *23*, 31, *39*
Davis, H. C., 564, 565, *571*, *572*
Davis, J. P., 192, *196*, 754, *758*, 1251, *1261*
Davis, K. E., 662, 673, *678*, 748, 770, *774*
Davis, L. G., 139, *146*
Davis, L. S., 85, *92*
Davis, M., 270, *294*, 331, *353*
Davis, R., 139, *146*

Davis, S., 565, *571*
Davis, T. J., 733, 735, *739*
Dawes, R., 502, *506*
Dawkes, H., 533, *543*
Day, L., 888, *892*
Day, M., 289, *293*
Dayton, T., 1046, *1059*, 1063, 1069, *1080*, *1081*
de Angeli, A., 394, 403, *411*, 649, 652, *655*, *656*
de Angelo, T., 403, *411*
De Bra, P., 570, *572*, 864, *867*
de Fockert, J. W., 54, *60*
de Glopper, K., 875, *883*
De Haes, H., 669, *677*
De Jong, R., 56, *59*
de Jong, T., 138, 139, *145*
de Labra, C., 50, *61*
de Paula, R., 646, *656*
de Ridder, S. N., 706, *717*
De Rose, T., 1033, *1038*
De Roure, D., 565, *571*
De Silva, G., 843, *850*
de Vreede, G.-J., 1022, *1038*
DeAngeli, A., 169, *175*, 420, *431*
Dearden, A., 729, *738*, *740*
Deary, I. J., 121, *131*
Deasy, C. M., 171, *173*
December, J., 604, *618*
Dechter, R., 580, *598*
Deci, E. L., 117, 122, 123, 129, *130*, *132*
Deeny, S. P., 1234, *1238*
Deering, M. J., 665, *676*
DeFanti, T. A., 517, *543*, 623, *636*
Deffenbacher, K. A., 44, *59*
Defty, N., 875, *883*
Deglise, P., 417, *431*
deGroot, A. D., 498, *506*
Dehdashti, P., 879, *881*
Deininger, R. L., 50, *59*
Deisigner, J., 628, *638*
Dekker, A. H., 610, *618*
DeKoven, E. A., 437*n*, 456
Delaney, C., 968, *988*
Delgado de la Flor, J., 1252, *1262*
delGaldo, E. M., 361, 376, 377, *377*, *378*, *379*
Della Sala, S., 219, *227*
Delta, H., 417, *429*, 840, *850*
Demasco, P. W., 818, *828*, 837, 840, 845, *850*, *851*, 862, *869*
DeMaso, D. R., 1256, *1261*
Dember, W. N., 45, *61*, 120, 127, *132*, 582, *598*
Demeure, A., 1118, *1124*
Demirdjian, D., 428, *428*
Den Boer, D.-J., 138, *146*
den Ouden, 491, *506*
Denckla, M., 876, *883*
Denecke, M., 415, *429*
Deninger, R. I., 29, *39*
Denning, P., 134, *145*
Denning, R., 583, *601*
Dennis, A. R., 15, *23*, 550, *557*
Dennis, F., 819, *828*
Dennis, I., 251, *263*
Dent, M., 567, *572*
Denz, M., 647, *658*
DePaula, R., 847, 848, *851*, 1077, *1078*
Derjani, 323, *325*
Derjani-Bayeh, A., 315, 323, *325*, *326*
DeRose, T., 169, *173*
DeSanctis, G., 15, *23*, 547, *555*
Desarkar, S., 580, *598*
Desmet, P., 1146, *1147*
Desurvire, H. W., 501, *506*, 754, *758*, 1141, *1147*, 1179, *1188*
Detenber, B. H., 79, *89*
DeTienne, K. B., 144, *145*
Detlefsen, E., 873, 875, 876, 877, 878, 879, *881*
Detmer, W. M., 668, *676*
Detterman, D. K., 813, 814, *827*
Deubel, H., 31, 33, *39*
Deutsch, D., 30, *39*, 54, *59*
Deutsch, J. A., 30, *39*, 54, *59*
Deutsch, S. E., 574, *601*
DeVellis, B. M., 138, *146*
Devine, M., 624, *637*
Devine, P. G., 770, *774*
DeVries, D., 746, *759*
deWaard, D., 708, *718*
Dewan, P., 647, *656*, *658*, 1249, *1262*
Dewey, D., 474, *485*
Dexter, A. S., 15, 18, *22*
Dey, A. K., 191, *196*, 642, 644, *656*, *657*, 1037, *1038*, 1255, *1261*
Dhamija, R., 647, 651, *656*
Dhillon, V., 33, *41*
Diaper, D., 575, *598*, 1202, 1205, *1207*
Dias, M. B., 591, *602*
Dibben, M., 583, *600*, 652, *657*
Dick, A. O., 1196, *1207*
Dick, G., 764, *775*
Dickhauser, O., 771, *774*
Dickinson, A., 826, *827*, 872, 876, 877, 879, *882*
Dickson, B. T., 1230, 1233, *1238*, *1239*
Dickson, G. W., 547, *555*, 673, *677*
Dickson, S., 877, *882*
Diday, E., 567, *572*
Dieberger, A., 1254, *1262*
Diehl, M., 1022, *1038*
Diehl, V. A., 71, 72, *75*, 87, *89*, 1138, *1147*
Diekmann, B., 1066, *1080*
Diened, B., 81, *89*
Diener, C. I., 771, *774*
Dietz, P. H., 171, *173*, *175*, 181, *196*
Dijk, 67, *76*
Dijkstra, J. J., 138, *145*
Dijkstra, S., 28, *39*
Dijkstra-Erikson, E., 1018, *1038*
Dillman, D. A., 972*n*, *987*, 1153, 1155, 1160, 1163, 1164, 1166, 1167, *1168*
Dillon, A., 217, *225*
Dillon, R. F., 162, *173*
Dimsdale, B., 531, *543*
Dindler, C., 799, *807*
Ding, X., 646, *656*
Ding, Z., 713, *719*
Dingawaney, A., 1064, *1078*
Dingledine, R., 643, *656*
Dingus, T. A., 702, 715, 715*t*, *716*, *717*
Dirkin, G. R., 118, *130*, *131*
diSessa, A. A., 804, *807*
Dittrich, Y., 1015, *1016*
Dix, A. J., 269, 270, 274, 275, 277, 278, 279, 286, 288, 289, *292*, *293*, 854, *867*

Dixon, M. A., 854, *868*, 1254, *1263*
DiZio, P., 628, 634, *636*, *637*, *638*
Doak, C., 873, 874, 876, *882*
Doak, L., 873, 874, 876, *882*
Dobson, S., 1108, 1119, *1124*
Docherty, P., 1063, *1078*
Dodge, M., 283, *292*
Doering, B., 1200, *1206*
Doherty, E., 837, 841, 849, *850*
Doherty, M., 579, *600*
Domik, G., 450, *456*
Domingos, P., 450, *458*
Donahue, G. M., 1276, *1277*
Donchin, E., 1227, *1239*
Donderi, D. C., 858, *867*
Donders, F. C., 45, 47, *59*
Donis, D. A., 334, *353*
Donker, A., 800, 801, *807*
Dooley, S., 1175, *1189*
Dorneich, M. C., 1228, *1238*
Dorris, A. L., 723, *738*
Dorval, R. K., 623, *636*
Douglas, J., 674, *677*
Douglas, S. A., 162, 163, 166, *173*, 914, *915*
Douglas-Cowie, E., 86, *89*
Douglass, L. W., 1234, *1238*
Douglass, S., 106, *111*, 575, *597*, 804, *807*
Dourish, P., 276, *292*, 551, 552, *555*, 642, 647, *656*, *657*, 795, *807*, 1030, *1038*, 1249, 1250, 1252, 1253, *1262*, *1263*, *1265*
Dours, D., 845, *849*
Dow, S., 1037, *1039*
Dowson, S. T., 541, *543*
Doyle, M., 1226, *1238*
Draper S. W., 754, *759*, 1020, *1039*
Draper, M. H., 630, *636*
Dray, S. M., 934, 938, 939, *946*, *947*, 1068, *1078*
Drew, H., 873, *881*
Drew, R., 843, *850*
Drews, F. A., 706, *716*
Drexler, J. M., 625, *637*
Dreyfuss, H., 993, *1015*
Driskell, J. E., 121, *130*, *132*
Driver, J., 30, 31, *39*, *41*, 56, *60*, 725, *740*
Droegemeier, K. K., 547, *555*
Drolet, A. L., 553, *555*
Droppleman, L. F., 87, *91*
Drost, L., 634, *636*
Drozd, A., 285, *292*
Drucker, S. M., 553, *556*
Druin, A., 795, 797, 798, 799, *806*, *807*, *809*, 1065, 1069, 1070, *1078*
Drummond, E., 32, *39*
Drury, C., 46, *59*, 218, *226*
Dryer, D. C., 139, *146*
Dua, U., 842, *851*
Dubberly, H., 1054, *1058*
DuBois, D., 581, *601*
Dubrovsky, V., 548, *557*
Ducheneaut, N., 548, *555*, 608, *618*
Duchnowski, P., 417, *430*
Duda, R. O., 624, *635*
Dudley, J. A., 729, *739*
Dudley-Sponaugle, A., 1254, *1264*
Dugas, C., 29, *40*
Duggan, G. B., 72, *75*
Dumais, S. T., 103, *112*, 172, *173*, 442, *457*, 496, *506*
Dumas, J. S., 751, *758*, 1130, 1132, 1133, 1134, 1135, 1136, 1137, 1139, 1140, 1141, *1147*, *1148*, *1149*
Duncan, B., 711*t*, *718*
Duncan, J., 31, *39*, 50, *59*, 218, *225*
Duncan, K. D., 150, *157*, 575, *597*, 1199, *1206*
Duncan, L., 415, 426, *429*, *431*, 863, *869*
Duncker, K., 81, *89*
Dundell, A., 765, *774*
Dunlap, R. D., 590, *598*
Dunlap, W. P., 629, 630, *637*
Dunn, D., 780, *790*
Dupé, G., 1121, *1124*
Dupont, S., 417, *429*
Duran, A., 576, *599*
Durfee, G., 648, 650, *655*
Durlach, B. N. I., 623, 626, *636*
DuRousseau, D. R., 1226, *1238*
Duvaisin, R., 833, *850*
Duval, E., 864, *869*
Dweck, C. S., 771, *774*
Dwivedi, P., 204, *226*
Dworkin, R., 1245, 1246, *1262*
Dyck, J. L., 278, *293*, 785, *790*
Dykes, P., 873, *881*
Dykstra, E. A., 1075, *1078*, *1080*, *1081*
Dyson, F., 3, *23*

E

Eades, P., 531, *542*
Earley, P., 365, *377*
Earp, J. B., 642, *655*
Eason, K. D., 560, *572*
Easterbrook, J. A., 118, *130*
Eberts, R. E., 55, *59*
Ebling, M., 1138, 1140, *1147*
Eby, D. W., 704, *716*
Eccles, R. G., 548, *557*
Echt, C. V., 785, *791*
Echt, K. V., 786, 791
Eckstein, M. P., 576, *598*
Eckstein, R., 1033, *1038*
Eday, J. D., 162, *173*
Ede, K., 1141, *1148*
Eden, H., 475, *486*
Edens, E., 583, *601*
Edmonds, E., 172, *175*
Edwards, A. D. N., 248, 257, 260, 260*f*, *263*, 837, *850*, 854, *867*
Edwards, C., 1253, 1255, *1261*
Edwards, E., 806, *807*
Edwards, K., 260, 261, *264*, 593, *600*
Edwards, N. E., 1255, *1261*
Edwards, W. K., 170, *172*, *196*, 283, *292*, 492, *506*, 863, *867*
Edworthy, J., 251, *263*
Efrat, A., 192, *196*
Efthimiou, E., 889, *893*
Egan, D. E., 788, *790*
Eggemeier, F. T., 120, *131*
Egger, F. N., 1254, *1262*
Egido, C., 548, 549, *555*, 1249, *1262*
Egly, R., 31, *39*

Ehn, P., 15, *22*, 1025, *1038*, 1062, 1063, 1065, 1066, 1069, 1071, 1072, 1073, 1075, *1077*, *1078*, 1250, *1262*
Ehret, B. D., 107, *111*
Ehrlich, S. F., 21, *23*, 551, *555*
Ehrlich, S. M., 87, *91*
Eick, S. G., 526, 531, 534, *542*
Eickmeier, B., 29, *40*
Eickstaedt, C., 1066, *1080*
Eilers, M. L., 780, *790*
Eimer, M., 47, *59*
Eisbach, C., 474, *486*
Eisdorfer, C., 782, *790*
Eisenberg, M., 804, *807*
Eisenstein, J., 1123, *1125*
Eisma, R., 872, 876, 879, *882*
Ekenel, H., 428, *429*
Ekin, A., 642, *658*
Eklof, M., 318, *325*
Ekman, P., 79, 85, 87, *89*, *90*, 420, *429*, 680, 681, 682, 683, 684*t*, 685, 686, 687, 688*f*, 690*f*, 691*f*, 692, 694*f*, *699*
El-Attar, T. E., 781, *790*
Elam, J. T., 858, *866*
Elashmawi, F., 365, *377*
Elder, L., 818, *828*, 838, 840, *849*
Elfstrom, G., 319, 320, *325*
Elias, M. F., 786, *790*
Elias, P. K., 786, *790*
Elias, R., 322, *325*
Elkin, E. H., 724, *738*
Elkind, J. I., 1195, *1207*
Elliot, A. M., 1250, *1263*
Elliot, D., 29, *39*
Elliot, K., 907, *915*
Elliot, R. G., 784, *790*
Elliott, C., 83, *89*
Elliott, D., 29, 33, 34, 35, 38, *39*, *40*, *41*
Elliott, J., 128, *130*, 806, *807*
Ellis, A., 492, *506*
Ellis, C., 22, 288, *294*, 546, *555*
Ellis, C. A., 288, *292*, 550, *555*
Ellis, E. R., 36, *41*, 735, 781, 785, *790*
Ellis, H. C. 81, *89*
Ellis, J. B., 262, *264*, 806, *807*
Ellis, J. T., 547, *556*
Ellis, L. B. M., 788, *790*
Ellis, S. R., 628, *638*, 735, *739*
Ellison, E., 628, *638*
Ellozy, H., 172, *176*
Ellsworth, P. C., 83, *89*
Elmore, W. K., 590, 591, *598*
Elrad, T., 1111, *1124*
Elrod, S., 550, *555*
Elson, S. A., 171, *174*
Elstein, A. S., 594, *598*
Embrey, D. E., 150, *157*
Emery, V. K., 861, *867*, *868*, *870*
Emigh, W., 606, *618*
Emiliani, P. L., 863, *867*, *870*
Emmelkamp, P. M. G., 634, *636*
Emmorey, K., 886, *892*
Encarnacao, L. M., 415, *429*
Enderwick, T., 574, 582, *600*
Endsley, M. R., 576, 583, *598*, 733, 734, *738*, *739*
Eng, P. K., 781, *791*
Eng, T. R., 665, *676*, *677*
Engel, R., 414, *431*
Engelbart, D., 5, *23*, 546, *555*
Engelbeck, G., 99, *112*
Engelhardt, Y., 521, *542*
Engelhart, D. C., 28, *39*
Engelien, B., 13, *23*
Engelland, S., 736, *740*
Engen, B. K., 1067, *1080*
Enger, C., 858, *869*
Engert, A., 711*t*, *718*
Engeström, Y., 615, *618*, 982, *986*
Englebart, D. C., 163, *173*
English, W. K., 28, *39*, 51, *59*, 163, 166, *173*, 546, *555*
Englmeier, K. H., 675, *677*
Engstrom, J., 708, *717*
Enns, J. T., 522, *542*
Enserink, B., 1022, *1038*
Entwistle, V., 669, *677*
Eoh, H., 706, *716*
Epps, J., 423, *429*
Erard, M., 886, *892*
Erber, M. W., 81, *89*
Erber, R., 81, *89*
Erdogmus, D., 1228, *1238*
Erenshteyn, R., 835, 840, *852*
Erickson, T., 796, *807*, 1046, 1049, *1058*, 1068, *1078*, 1255, *1263*
Ericsson, E., 581, *597*
Ericsson, K. A., 64, *75*, 121, *130*, 798, 800, 801, *807*, 1139, 1140, *1147*
Eriksen, B. A., 31, *39*, 725, *738*
Eriksen, C. W., 31, *39*, 725, *738*
Eriksson, E., 799, *807*
Erkens, G., 596, *598*
Ernst, M., 422, 428, *429*
Ervin, R. D., 702, *719*
Erzberger, H., 732, 736, 737, *738*
Escudier, P., 415, *431*
Esemplare, M., 1176, *1189*
Espinosa, J. A., 73, 74, *75*
Esposito, C., 415, *429*
Essa, I. A., 85, *90*, 181, *196*
Eun-Young, K., 139, *146*
Evanoff, R., 1064, *1078*
Evans, D. G., 843, *850*
Evenson, S., 17, *23*
Everitt, K., 171, *176*
Eversole, A., 168, *172*
Eysenck, H. J., 125, 127, *130*
Eysenck, M. W., 125, 127, 128, *130*

F

Fabiani, M., 1230, *1238*
Fabre, J., 274, *293*
Fabricant, R., 695, *699*
Fadden, S., 730, *740*
Faden, R., 1255, *1262*
Fagerjord, A., 569, *572*
Fahlen, L., 277, *292*
Fahlen, L. E., 804, *808*
Fahy, P. J., 610, *618*
Fails, J. A., 172, *173*, 181, *196*, 799, *807*
Fairchild, K. M., 531, *542*

Fairweather, P., 796, *808*
Faisman, A., 891, *892*
Fait, H., 1254, *1264*
Falcão V., 172, *176*, 180, *198*, 643, *658*
Falconer, S., 128, *131*
Falk, J., 285, *292*
Falk, R., 502, *506*
Fanderclai, T., 1065, *1078*
Faneuf, R., 1119, *1124*
Fang, G., 890, *892*
Fang, N., 142, *145*, *279*, *292*, 652, *656*
Fano, R., 4, *23*
Faraday, P., 394, 401, 408, *410*, *411*
Faraj, S., 652, *657*
Farber, A., 799, *807*
Farber, E., 713, *716*, 732, *738*
Farina, F., 765, *774*
Farley, T., 736, *739*
Farneuf, R. 1119, *1124*
Farnham, S. D., 553, *556*
Farooq, U., 1065, *1080*
Farquhar, B. B., 6, *24*
Farrell, R., 501, *506*
Farrell, S., 1255, *1262*
Farstrup, A., 217, *226*
Fasel, I., 451, *455*
Faulkner, L., 1140, *1147*
Faulring, A., 1035, *1039*
Faust, D., 502, *506*
Favalora, G. E., 623, *636*
Favart, M., 69, *76*
Favio, R-A., 765, *774*
Favre, J. M., 1116, *1124*
Fawcus, R., 835, 840, *852*
Fay, E. A., 891, *892*
Fayard, A.-L., 1018, *1039*, 1250, *1264*
Faye, E., 859, *869*
Fearing, R. S., 233, *245*
Feehrer, C. E., 1195, *1207*
Fehling, M., 451, *456*
Feigenbaum, E. A., 13, *23*
Feil, N., 820, *827*
Feiner, S., 310, *311*, 518, 534, *542*, 576, *597*
Feiner, S. K., 204, *225*
Feinsod, M., 856, *868*
Fekete, J.-D., 1033, *1038*
Feld, P., 579, *602*
Felder, R. M., 674, *676*
Feldman B. L., 87, *90*
Feldman, E., 1251, *1263*
Feldman, S. I., 547, *555*
Felici, J., 335, *353*
Felix, C. P., 502, *507*
Fell, H. J., 417, *429*, 840, *850*
Fellenz, W., 85, *89*
Felten, E., 279, *293*, 1244, 1251, 1255, *1262*, *1264*
Feltovich, P., 581, *597*, *599*
Felzer, T., 842, *850*
Fencott, C., 628, *636*
Feng, J., 190, *196*, 384, *390*, *391*, 640, 642, 645, 657, 835, 838, 844, 847, *850*, *851*, *852*
Fenwick, J., 569, *572*
Ferguson, T., 665, 666, *676*, 781, *791*
Fernandes, T., 361, 377, *377*, 795, *807*
Fernandez, R., 439, *456*
Ferrell, W. R., 581, *601*, 728, *738*
Ferrency, A. S., 1035, *1039*
Ferrero, S., 298, *312*
Ferrier, L. J., 417, *429*, 840, *850*
Ferris, P., 605, *618*
Ferriz, J., 288, *294*
Ferro, D., 863, *869*
Fertig, S., 526, *542*
Feuer, W. J., 858, *869*
Fiegel, T., 1180, *1188*
Fields, B., 729, *740*
Figueroa, P., 633, *636*
Filho, R., 646, *656*
Findlater, L., 453*f*, *456*, 879, *882*
Findlay, J. M., 31, *40*
Finholt, T. A., 547, 552, 553, 554, *555*, *556*, *557*, 643, *656*
Finlay, J., 269, *292*, 854, *867*
Finn, K., 553, *555*
Finn, R., 217, *225*
Fiore, S., 583, *601*
Fischer, G., 406, *410*, 570, *572*, 593, *598*
Fischer, M., 859, *869*, 968, *987*
Fischer-Huebner, S., 644, *657*
Fischler, M. A., 184, *196*
Fiset, V., 669, *677*
Fish, R. S., 552, *555*
Fisher, G., 475, *486*
Fishkin, J. S., 1245, *1262*
Fishkin, K. P., 170, 172, *174*, *176*, 182, *198*, 474, *486*
Fishwick, L., 652, *658*
Fisk, A. D., 778, 784, 786, *790*, *791*
Fiske, S. T., 83, *92*
Fitchett, C., 178, *196*, 1037, *1038*
Fitton, D., 449, *455*
Fitts, P., 166, 166*f*, *173*
Fitts, P. M., 28, 29, *39*, 44, 50, 56, *59*, 722, 723, 724, 725, 726, 727, 737, *738*
Fitzmaurice, G. W., 163, 164, 167, 169, 171, 172, *172*, *173*, *174*, *175*, 180, 194, *196*, 484, 485, *486*, 1032, *1039*
Fitzpatrick, L., 567, *572*
Fitzpatrick, S. J., 137, 138, *146*
Fjeld, M., 475, *487*
Flach, J. M., 121, 123, *130*, 576, 581, *597*, *598*, *599*, 734, *738*
Flaherty, G., 428, *429*
Flake, G. W., 564, *572*
Flanagan, J. C., 415, 417, *429*, *432*, 943, *946*
Flanagan, M., 1251, *1262*
Flanagin, A. J., 142, *146*
Flanders, A., 1184, *1189*
Flash, T., 166, *176*
Flechais, I., 647, *657*
Fleming, M. W., 644, *655*
Flesch, R. F., 503, *506*
Fletcher, C. R., 67, *75*
Fletcher, D. C., 855, 858, *867*
Fletcher, H. R., 475, *486*
Fletcher, K., 1233, *1238*
Flicker, M., 181, *196*
Flickner, M., 181, *196*, 414, 417, *430*, *431*
Flieschman, 1137, *1149*
Flintham, M., 285, *292*
Florencio, D., 180, *198*
Flowers, J. H., 261, *264*
Floyd, C., 1062, 1065, 1067, *1078*, 1250, *1262*

Floyer-Lea, A., 1234, *1238*
Fluehr-Lobban, C., 968*n*, *986*
Flynn, M., 703, *717*
Fodor, I. K., 185, *196*
Foehr, U., 87, *91*
Fogarty, J., 439, *456*, 1255, *1262*
Fogel, D., 580, *600*
Fogel, L., 726, *738*
Fogg, B., 474, *486*
Fogg, B. J., 82, *90*, 134, 136, 137, 138, 139, 140, 141, 142, 143, 144, *145*, *146*, 279, *292*, 398, 404, *410*, 474, *486*, 652, *656*, 1254, *1262*
Foley, J., 1213, *1218*
Foley, J. D., 7, *23*, 110, *111*, 165, *173*
Foley, J. P., 713, *716*
Folk, C. L., 31, 32, 34, *39*
Folkman, S., 116, 117, *131*
Follert, R. L., 723, *738*
Folstein, M. F., 858, *869*
Fong, M. W., 415, *431*
Foot, P., 1245, *1262*
Forbus, K. D., 19, *23*
Ford, M. E., 137, *145*
Forgas, J. P., 81, *89*, *90*
Forlines, C., 194, *198*
Forlizzi, J., 1255, *1262*
Forrest, J., 1167, *1169*
Forsyth, D. A., 179, *196*, 576, 579, 583, *598*
Forte, A., 552, *555*
Fortner, R. S., 604, *618*
Fortuin, F. T., 854, 865, *867*
Foster, G., 276, *294*
Foster, I., 269, *292*, 547, *555*
Foster, J., 647, *658*
Fotinea, S.-E., 889, *893*
Fouad, H., 624, *636*
Foucault, B., 943, 944, 946, *947*
Foulds, R., 835, 840, *852*
Fourer, R., 581, *600*
Fowles, R. A., 1062, 1065, 1066, 1067, *1078*
Fowlkes, J. E., 629, *637*
Fox, A., 1123, *1125*
Fox, D., 179, *197*
Fox, S., 666, *676*
Foxlin, E., 625, *636*
Foyle, D. C., 732, *739*
Fozard, J. L., 492, *506*, *507*, 784, *791*
Fracker, M. L., 734, *738*
Fraden, J., 178, 179, *196*
Francis, A., 385, *390*
Francis, G., 725, *738*
Francis, W. N., 492, *506*
Frank, M. P., 415, *431*
Frank, T., 796, *807*
Frankel, J. L., 474, *485*
Frankel, M. S., 1259, *1262*
Frankowski, D., 442, *457*, 553, *555*
Franks, J. J., 53, *60*, 67, *74*, 496, *505*
Franzen, S., 702, *718*
Franzke, M., 103, *111*
Fraser, D., 854, *808*
Fraser, J., 864, *867*, 913, *915*
Fraser, J. M., 578, 595, *598*
Fraser, L., 495, *506*
Fraser, M., 278, *293*, 554, *556*
Frazer, J., 474, *486*
Frazer, P., 474, *486*
Frederking, R. E., 308, *311*
Freed, A. R., 108, *113*, 575, *601*
Freed, J., 993, *1015*
Freed, M., 1198, *1207*
Freeman, E., 526, *542*
Freeman, F. G., 120, *132*, 582, *598*, *600*, 1227, *1238*, *1239*
Freeman, J. M., 858, *866*
Freeman, W. T., 170, *173*
Frei, P., 473, 474, 476, *486*
Freid, A. N., 855, *869*
Freidman, J. H., 186, *196*
Freier, N. G., 1251, 1255, *1262*, *1263*, *1266*
Freisleben, B., 842, *850*
Freitag, D., 436, *457*
French, C. N., 1234, *1238*
Freud, S., 1044, *1058*
Freudenthal, A., 814, *827*
Freudenthal, D., 787, *791*
Freuder, E., 581, *600*
Freyne, J., 442, *457*
Fricke, J., 862, *868*
Friday, A., 277, *292*
Fridlund, A., 420, *429*
Friederici, A. D., 1234, *1238*
Friedlander, C., 863, *868*
Friedman, A., 15, *23*
Friedman, B., 83, *90*, 140, *145*, 279, *293*, 653, *656*, 1243, 1244, 1247, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1255, *1261*, *1262*, *1263*, *1264*
Friedman, D., 705, *718*
Friedman, T., 1218, *1219*
Friesen, W. V., 85, 87, *89*, *90*, 420, *429*
Friestad, M., 81, *92*, 682, *700*
Frijda, N. H., 79, 80, 84, 85, *90*
Frishberg, N., 888, 889, *892*, 1025, *1038*
Frisse, E. H., 668, *676*
Fritsch, C., 841, *851*
Frobert, L., 1031, *1039*
Froggatt, K. L, 1077, *1078*
Frøkjær, E., 1138, *1147*, 1185, 1186, *1188*
Fromm, E., 122, *130*
Fröst, P., 1073, *1079*
Fruehauf, N., 204, *225*
Frysinger, S. P., 251, *264*
Fu, W., 108, *112*
Fu, X., 549, *557*
Fuchs, L., 1249, *1262*
Fuchs, M., 711, *718*
Fuchs-Kittowski, K., 711*t*, *718*, 1063, *1078*
Fuhrer, M. J., 849, *850*
Fujii, T., 231, 240, *244*
Fukumoto, M., 426, *429*
Fukunda, T., 220, *227*
Fuller, R., 439*n*, *456*
Fuller, T., 1141, 1145, *1148*
Fulton, B., 754, 749, 748, 750, *759*, 1138, 1146, *1148*
Fulton, J., 137, *146*
Funge, J., 742, *758*
Funkhouser, T., 171, *173*
Furlong, M. S., 780, *791*
Furnas, G. W., 496, *506*, 531, 537, *542*, 563, *572*
Furuta, K., 1138, *1149*

Fussell, S. R., 548, 549, 550, *555*, *556*
Fuster-Duran, A., 424, *429*

G

Gaba, D. M., 1136, *1148*
Gabbard, J. L., 630, *636*
Gabriel, G. C., 1065, *1079*
Gabriel, L. T., 813, *827*
Gabrielli, E., 647, *655*
Gabrys, G., 877, *881*
Gage, P., 786, *790*
Gagliardi, A., 670, *677*
Gagne, M., 122, 123, 129, *130*
Gagné, R. M., 605, *619*
Gahm, G. A., 141, *145*
Gaillard, A. W. K., 116, *131*
Galanter, E., 575, *600*
Gale, C., 549, *555*, *556*
Gale, M. T., 764, 765, *774*
Galegher, J., 1249, *1262*
Galer, M., 723, *740*
Gallagher, J. P., 715, *716*
Galletta, D., 15, *24*
Gallienne, R. L., 782, *791*
Gallini, J. K., 70, *75*
Gallupe, R. B., 15, *23*
Galotti, K. M., 612, *618*
Galvankar, P., 874, *882*
Galvin, J. C., 846, *852*
Galvin, R. W., 662, *676*
Galyean, T., 192, *196*
Gamm, S., 318, *325*
Gamma, E., 1033, *1038*
Gandy, M., 1037, *1039*
Ganeshan, R., 879, *882*
Ganoe, C. H., 1065, *1080*
Gao, J., 1054, *1058*
Gao, W., 890, *892*, *893*
Garavan, H., 576, *598*
Garay-Vitoria, N., 837, 845, *850*
Garb, H., 577, *598*
Garcia-Molina, H., 171, *173*, 547, *555*
Gardano, A. C., 84, *90*
Gardiner, M., 395, *410*
Gardner, R. M., 376, *377*, 667, *676*
Garfield, E., 567, *572*
Garfinkel, S., 182, *196*, 1244, 1245, 1252, *1262*
Garg, A., 415, 417, *431*, 833, *851*
Gargan, R. A., 415, 424, 426, *429*
Gargiulo, R., 171, *175*
Garlan, D., 1121, *1124*
Garland, A., 437, *457*
Garland, D., 576, *598*
Garren, J. F., Jr., 725, *737*
Garrett, J. J., 898, 899*f*, 903, 905, *915*
Garrety, K., 1077, *1079*
Garrison, R. D., 610, *618*
Gärtner, J., 1063, *1079*
Garton, L., 610, 611, *618*
Garver, J. A., 764, *774*
Garzotto, F., 570, *571*
Gash, D. C., 552, *557*
Gasparetti, F., 442, *457*
Gasper, K., 80, *89*, 682, *699*
Gasser, M., 647, *655*
Gasson, S., 1077, *1079*
Gatica-Perez, D., 428, *429*, *430*
Gatignon, H., 141, *145*
Gauch, S., 442, *457*
Gaver, W. W., 248, 249, 255, 255*f*, 256, 259, *263*, *264*, 276, *292*, 854, 863, *868*, 1036, *1038*
Gay, L. R., 803, *807*
Gebotys, R. J., 138, *146*
Gedde, S. J., 858, *869*
Geddes, N. D., 576, 580, 581, 582, 583, 590, 591, *598*, *601*
Gee, J., 692, *699*
Geelhoed, E., 549, *558*
Gehrke, H., 706, *718*
Geibler, J., 1112, *1125*
Geiger, D., 187, *197*
Geister, S., 583, *598*
Gelade, G., 30, *41*, 55, *61*
Gelau, C., 710, *718*
Gelfand, S. A., 251, 252, *264*
Geller, E., 667, *676*
Gellersen, H.-W., 182, *198*, 277, *293*
Gelpi, A., 769, *774*
Gemperle, F., 298, *311*
Gena, C., 451*n*, *456*
Gendall, P., 1167, *1168*
Gentner, D., 64, *74*, *75*, 576, *598*, 731, *738*
Georgas, J., 137, *145*
George, J., 15, *23*, 550, *557*
George, M., 138, *146*
George, P., 1053, *1058*
Gerard, R., 84, *90*
Gerber, B., 873, *881*
Gerber, E., 854, *868*
Gerber, P., 874, *882*
Gergle, D., 548, 553, *555*, *556*, *558*, 1254, *1261*
Gerrity, M., 674, *677*
Gershon, J., 634, *636*
Gerson, A., 1227, *1239*
Gervasio, M. T., 436, *456*
Gettlefinger, T. C., 858, *866*
Gevins, A., 1223, 1234, *1238*, *1239*
Gewirth, A., 1245, *1262*
Ghallib, M., 581, *598*
Ghanayim, N., 841, 849, *851*
Ghiringhelli, L., 321, *325*
Ghorbani, A. A., 652, *658*
Ghose, K., 862, *869*
Ghosh, A., 139, *145*
Giaccardi E., 406, *410*
Giacomi, J., 966, *986*, 1066, *1077*
Giangola, J., 382, *390*
Gibbons, A., 579, *602*
Gibbons, J., 172, *176*, 180, *198*, 643, *658*
Gibbs, S. J., 288, *292*, 550, *555*
Gibbs, W. W., 217, *225*
Gibson, J. J., 57, *59*, 123, *130*, 167, *173*
Gibson, S., 283, *293*
Giffin, W., 578, *601*
Gilbert, D. T., 83, *92*
Gilbert, G. A., 730, *738*
Gilbertson, L., 835, 843, 847, *851*

Giles, J., 552, *555*, 564, *572*
Gill, B., 1251, 1253, *1262*
Gill, J., 825, *827*
Gill, R. T., 575, 576, 577, *598*
Gillan, D. J., 51, *59*, 861, *868*
Gilles, C., 1167, *1168*
Gilligan, S. G., 81, *89*
Gilliland, K., 128, *131*
Gilmartin, K. J., 498, *507*
Gilmore, D., 1133, *1149*
Gilmore, J., 976, *987*
Gilmore, J. H., 904, *915*
Gilovich, T., 578, 595, *598*
Gilutz, S., 795, 797, *807*
Giovinco, M. G., 623, *636*
Girand, C., 422, *432*
Gist, M., 786, *791*
Giuse, D. A., 1035, *1039*
Gjersvik, R., 1065, *1079*
Gladwell, M., 500, *506*
Glance, N., 550, *555*
Glas, D., 194, *198*, 474, *487*
Glaser, R., 580, *599*
Glenberg, A. M., 68, 69, *75*
Glendon, A. I., 154, *157*, 786, *792*
Glenn, F. A., 1196, *1207*
Glinert, E. P., 865, *868*, 1102, *1105*
Glock, M. D., 403, *410*
Glover, E., 550, *558*
Gnanayutham, P., 842, *850*
Goble, J. C., 162, *174*, 854, 865, *868*
Goddeau, D., 414, *432*
Godden, D. R., 53, *59*
Godefroid, P., 552, *555*, 1253, *1263*
Godjin, R., 38, *39*
Goette, T., 845, 847, *850*
Gohm, C., 81, *89*
Gold, E., 1055, *1059*
Gold, R., 1036, *1039*
Gold, W. R., 138, *145*
Goldberg, A., 7, *23*, 472, *486*, 1055, *1059*
Goldberg, D., 169, *173*, 1254, *1263*
Goldberg, J. H., 1139, *1148*
Goldberg, K., 1252, *1263*
Goldberg, S. L., 625, *637*
Golden, E., 439, *456*
Goldman, R., 384, *391*, 574, *600*, 835, 844, *850*, *851*
Goldman-Segall, R., 797, *807*
Goldsmith, J., 863, *868*
Goldstein, B., 834, *852*
Goldstein, L., 574, *600*
Goldstein, M. K., 224, *225*, 667, *676*
Goldwaithe, J., 842, 849, *851*
Goleman, D., 88, *90*
Golovchinsky, G., 172, *176*
Gomez, L. M., 496, *506*, 788, *790*, 1249, *1265*
Gong, L., 87, *91*
Gonzalez, V. M., 492, *506*
González-Abascal, J., 837, 845, *850*
Gonzalez-Heydrich, J., 1256, *1261*
Good, C., 770, *773*
Good, L., 1033, *1038*
Good, M. D., 1026, *1038*, 1053, *1058*
Good, N. S., 642, 644, *656*, 1253, *1263*
Goodall, J., 651, *656*
Goodeve, P. J., 29, *39*
Goodman, D., 28, 29, *39*, *40*, 1030, *1038*
Goodman, M. J., 704*t*, *719*
Goodwin, G. F., 72, *75*
Goodwin, N. C., 6, *24*
Gopher, D., 123, 124, 125, *131*, 633, *636*
Goraczko, M., 180, *198*
Gorbet, M., 284, *294*, 474, *486*
Gordin, D., 501, *506*
Gordon, A., 581, *598*
Gordon, L., 168, *175*
Gordon, S. E., 574, 575, 576, 577, *598*
Gordon-Salant, S., 784, *791*
Goren-Bar, D., 454, *456*
Gorman, A., 1077, *1078*
Gorniak, P., 428, *429*
Goshen-Gottstein, Y., 51, *61*
Gosling, G. D., 734, *738*
Gossweiler, R., 171, *176*
Gotterbarn, D., 1257, *1261*, *1263*
Goulandris, N., 875, *883*
Gould, E. W., 377, *377*, 378, *379*
Gould, J. D., 217, 219, 221, *225*, 497, 503, *506*, *507*, 702, *716*
Gould, S. J., 813, *827*
Goulding, J., 567, *572*
Gouldner, A. W., 140, *145*
Graber, M. A., 875, *882*
Grabowski, M., 491, *506*, 574, *598*
Gracio, D., 1074, *1078*
Grady, C. L., 51, *61*
Graeber, D. A., 628, *638*
Graf, M., 15, *22*
Graf, P., 53, *61*, 879, *882*
Gragoudas, E. S., 856, *868*
Graham, D., 727, *739*
Graham, T., 359, *379*
Gram, I. T., 1155, *1169*
Gramopadhye, A., 218, *225*
Grandjean, E., 6, *23*, 314, 315, 316, 321, *325*
Grant, A. E., 610, *619*, 649, *658*
Grant, E. C., 420, *430*
Grant, K. D., 171, *176*
Grant, K. R., 548, *556*
Grasso, A., 501, *506*
Grasso, F., 444, *455*
Gratton, G., 1230, *1238*
Gravier, G., 415, 417, *431*
Gray, D. B., 837, *850*
Gray, J. A., 127, 128, *130*, 781, *790*
Gray, M. J., 150, *157*, 575, *597*, 1199, *1206*, *1207*
Gray, P. D., 168, *175*, 259, *263*, 274, *293*, 839, *851*, 859, *869*
Gray, W. D., 12, *23*, 94, 95, 105, 109, *111*, *112*, 286, *293*, 490, *506*, 929, *947*, *1043*, 1053, *1058*, *1059*, 1179, 1180, 1186, 1187, *1188*, 1198, 1199, 1203, *1207*
Graybiel, A., 628, *637*
Grayson, D. M., 549, *555*
Graziola, I., 454, *456*
Gredler, M. B., 138, *145*
Green, P. A., 563, *572*, 702, 703*t*, 705, 706, 707, 708, 709*t*, 710, 711, 711*t*, 712, 713, 714, *716*, *717*, *718*, *719*, 723, *738*
Green, R. A., 323, *325*
Green, S., 969*n*, 975, *986*
Green, T. G. R., 10, *24*, 502, 503, *505*, *506*, *947*, 1176, 1177, *1188*

Greenbaum, J., 6, *23*, 1021, *1038*, 1062, 1063, 1076, 1077, *1079*, *1080*, *1081*, 1250, *1263*
Greenbaum, T. L., 757, *758*
Greenberg, J., 713, *716*
Greenberg, R., 248, *263*, 863, *867*
Greenberg, S., 2, *22*, 178, 194, *196*, *197*, 287, *294*, 552, *556*, 576, *597*, 647, *656*, 907, *915*, 1033, 1037, *1038*, *1039*, 1152, *1168*, 1253, *1261*
Greene, D., 746, *759*
Greene, S. L., 499, *506*
Greenes, R. A., 667, *676*
Greenfield, P., 772, *775*
Greenhalgh, C., 278, *292*, *293*, 554, *555*, *556*
Greenidge, K., 1254, *1264*
Greenspan, S., 1254, *1263*
Greenwood, P., 582, *599*
Greer, J. D., 142, *145*
Gregor, P., 819, 823, 824, 825, 826, 827, *827*, *828*, 872, 876, 879, *882*
Gregoriades, A., 1055, *1058*
Gregory, A. H., 84, *90*
Gregory, D. F., 668, *677*
Greif, I., 15, *23*, *555*
Grenfell, M., 1064, *1079*
Gress, W., 632, *636*
Gribble, S., 797, *807*
Gribbons, W., 872, *882*
Grief, I., 546, *555*, 966, *986*, 1249, *1263*
Grier, R., 582, *598*
Griesemer, J. R., 736, *740*, 1004, *1016*, 1070, *1081*
Griffin, D., 578, *598*
Griffin, P., 802, *808*
Griffith, D. C., 973, *987*
Griffith, E. R., 832, *851*
Griffith, T. L., 583, *598*
Griffiths, V., 876, *881*
Grinter, R. E., 170, *172*, *196*, 492, *506*, 548, 552, *555*, *556*, 648, *655*, 1252, *1262*
Grintner, R. E., 986*n*, *987*
Grischkowsky, N., 217, *225*
Gritz, L., 624, *636*
Grobelny, J., 46, *59*
Groeneweg, J., 148, *158*
Gröger, T., 219, 220, 222, *225*, *226*, *227*
Grønbæk, K., 1066, 1072, 1075, *1078*, *1079*
Grondin, S., 576, *598*
Gross, C. R., 781, *790*
Gross, J. J., 78, 80, *90*
Gross, M. D., 1250, *1263*
Grossklags, J., 643, *658*
Grossman, T., 168, 170, *174*
Grossnickle, M. M., 1140, *1148*
Grudin, J., 2, 6, 18, 20, 21, 22, *22*, *23*, *24*, 140, *145*, 171, *174*, 546, 551, *555*, *556*, *557*, 576, *597*, 978, 979, 980, 986*n*, *987*, 992, 993, *1015*, 1152, *1168*, 1242, 1248, 1249, 1253, *1263*, *1265*
Gruen, D., 435*n*, *456*, 1069, 1070, *1079*
Grunnet, C., 1067, 1069, 1071, 1072, *1077*
Guan, Z., 165, *175*
Gubrium, J. F., 971, *987*
Guerlain, S., 583, 594, 595, *598*, *601*
Guerrier, J. H., 780, 785, 787, *790*
Guest, G., 975, *987*
Guggiana, V., 862, *867*
Guha, M. L., 799, *807*
Guiard, Y., 51, *59*, 163, 166, 169, *174*, *175*, 185, *196*
Guiard-Marigny, T., 420, *428*
Guimbretiere, F., 165, 171, 172, *172*, *174*, *175*
Guiney, E., 574, *599*
Gujar, A., 170, 172, *174*, *176*, 182, *198*
Gulli, A., 561, 567, *572*
Gullo, D. F., 804, *807*
Gumbrecht, M., 550, *557*
Gunawardena, C., 608, 610, *618*
Gundel, A., 1223, *1238*
Gunn, D., 748, *759*
Gunnarsson, E., 314, 321, *325*
Gunning, R., 503, *506*
Gunter, T. C., 1234, *1238*
Gunther, E., 474, *486*
Gupta, A., 428, *430*, 549, *555*, 1253, *1263*
Gurevitch, M., 88, *90*
Guruprasad, S., 280, *292*
Gustafson, D. H., 665, *676*, *677*, 781, *791*
Gustafsson, E., 318, *325*
Gutkauf, B., 450, *456*
Guttmann, P., 650, *656*
Gutwin, C., 278, *293*, 552, *556*, 854, 864, *867*
Guy, T., 1140, 1143, *1149*
Guyer, D. R., 856, *868*
Guyomard, M., 414, *432*
Guzdial, M., 802, 804, 806, *807*, *808*
Gyoba, J., 85, *90*

H

Ha, Y., 579, *599*
Haag, Z., 765, *774*
Habberg, M., 318, *325*
Haber, E., 648, *657*
Habermas, J., 1246, *1263*
Hacisalihzade, S., 219, *226*
Hacker, E. W., 733, *738*
Hackett, J. P., 977, *987*
Hackman, G. S., 1136, *1148*
Hackman, R. J., 1152, *1168*
Hackos, J. T., 909, *916*, 922, *926*, 935, 936, 940, *947*, 1276, *1277*
Hadar, U., 420, *430*
Hadley, G., 1227, *1239*
Haeb-Umback, 318, *325*
Haeckel, S., 911, *915*
Hagan, P. R., 653, *656*
Hagberg, M., 314, 318, 323, *325*
Hagenauer, M. E., 677
Hagerhall, C. M., 491, *507*
Haggerty, B., 318, *325*
Hagman, J., 1251, *1262*, *1263*
Hagström, A. E., 277, *293*
Hahn, J. K., 624, *636*
Haims, M. C., 314, *324*
Hair, D. C., 1176, *1189*
Hajdukiewicz, J., 577, 580, *597*
Hakkani-Tür, D., 440, *457*
Halasz, F. G., 65, 66, 68, *75*, 274, *293*, 550, *555*
Hale, K. S., 624, 626, 633, *636*, *637*, *638*
Halgren, S., 795, 796, *807*
Hall, C., 646, *657*
Hall, D. M., 365, *379*, 623, *636*
Hall, E. T., 553, 554, *556*
Hall, J., 768, *774*

Hall, L. K., 53, *59*, 1181, 1182, 1185, *1188*
Hall, M., 262, *263*, 285, *292*
Hall, R., 564, 565, *556*
Hall, W., 407, *411*, 565, *571*, *572*
Halle, M., 623, *636*
Haller, S., 439, *456*
Hallett, M., 834, *850*
Hallewell, H. J. D., 1063, *1080*
Halskov M. K., 1077, *1080*
Halstead, M. H., 493, 495, 502, 503, *506*
Halverson, C. A., 169, *174*, 190, *197*, 384, *391*
Halvorsen, R. F., 606, *618*
Hamalainen, P., 800, *807*
Hamid, R., 1037, *1038*
Hamilton, H., 890, *893*
Hamilton, J. A., 1123, *1125*
Hamilton, P., 116, *131*
Hamilton, S., 890, *893*
Hammer, J. M., 590, 591, *598*, *601*
Hammond, K. J., 442*n*, *455*, 785, 786, 788, *790*
Hammond, M., 834, *852*
Hammond, N. V., 442, *456*, 1181, *1188*
Hampapur, A., 642, *658*
Han, J. Y., 171, *174*, 181, *196*
Han, R., 1123, *1124*
Han, Y. Y., 667, *676*
Hancock, P. A., 116, 117, 117*f*, 118, 118*f*, 119*f*, 120, 121, 122, 123, 125, 126, 127, 128, 128*f*, 129, *130*, *131*, *132*, 581, *598*, 729, 734, *739*, *740*, 1145, *1148*
Handberg, R., 578, *598*
Handel, M., 643, *656*
Handy, C., 583, *598*
Hankey, J. M., 715, 715*t*, *717*
Hanks, S., 448, *456*, 581, *600*
Hanna, G. B., 37, *39*
Hanna, L., 796, 797, 800, *807*, 1143, *1148*
Hanneman, R. A., 610, 611, *618*
Hannum, W., 575, *599*
Hanowski, R. J., 715, 715*t*, *717*
Hanrahan, P., 1123, *1125*
Hansen, J., 843, *850*
Hansen, P. K., 846, *852*
Hansman, R. J., 734, 736, *739*
Hanson, A., 205, 211*f*, *226*
Hanson, E., 162, *174*
Hanson, R. H., 724, *738*
Hanson, V. L., 436, *456*, 504, *506*, *507*, 887, 888, *892*, *893*
Hansson, R., 285, *292*
Harackiewicz, J. M., 81, *90*
Harbusch, K., 414, *430*
Harder, K., 250, *263*
Hardin, J. B., 547, *557*
Harding, S., 968, *987*
Hardouin, J., 51, *59*
Hardstone, G., 583, *597*
Hariharan, R., 192, *196*
Harkreader, A., 837, 844, *851*
Harman, D., 561, 566, 567, *572*
Haro, A., 181, *196*
Harootunian, H., 1076, *1078*
Harper, H. R., 172, *176*
Harper, J., 643, *656*
Harper, R., 277, *294*, 733, *738*, 908, *915*, 982, *986*
Harper, S., 854, 865, *868*
Harpold, T., 171, *174*
Harrington, D. L., 50, *60*
Harris, J., 365, *377*, 665, *676*
Harris, P. R., 365, *377*, 652, *658*, 665
Harris, R. M., 1196, *1207*
Harris, S. C., 118, *131*, 170, *174*, 188, *197*, 595, *598*
Harris, T. A., 764, 765, *774*
Harris, W. C., 118, *131*
Harrison, B. L., 170, 171, 172, *174*, *176*, 182, *198*
Harrison, M., 729, *738*
Harrison, S., 551, *556*, 1250, *1263*
Hart, D. J., 6, *22*, 207, *225*, 314, *324*
Hart, R. P., 503, *506*
Hart, S. G., 120, 127, *131*
Hartson, H. R., 1086, *1105*, 1141, *1147*, 1177, 1178, 1185, 1186, *1188*
Hartson, R., 1108, *1124*
Harvey, P., 975, *986*
Hasbroucq, T., 48, 49, 51, *59*, *60*
Haselton, M. G., 578, 595, *598*
Hasling, D. W., 583, *598*
Hassebrock, F., 576, *599*
Hassenzahl, M., 684*t*, *699*, 899*n*, *915*, 1146, *1148*
Hassmen, P., 139, *145*
Hastie, R., 578, *600*
Hatch, E., 1246, *1263*
Hatfield, B. D., 1234, *1238*
Hatfield, E., 83, *90*
Haubner, M., 675, *677*
Haubner, P., 220, *226*
Hauer, T. A., 261, *264*
Haumer, P., 1046, *1059*
Hauptmann, A. G., 170, *174*, 419, *430*, *431*
Haviland, S. E., 1044, *1058*
Hawley, P., 331, *353*
Haworth, C., 705, *718*
Hay, J. L., 139, *145*
Hay, K. E., 617, *618*, 802, *808*
Hayek, J. H., 1167, *1168*
Hayes, B. C., 574, 576, 583, 596, *599*, 787, *792*
Hayes, J. R., 218, *227*, 582, *599*
Hayes, K. H., 781, 787, *790*
Hayes, P J., 1123, *1124*
Hayes-Roth, B., 87, *90*, 581, *598*
Hayes-Roth, F., 581, *598*
Haynes, S. R., 1055, *1058*
Hays, R. T., 633, *636*
Haythornthwaite, C., 610, 611, *618*
Hazen, M., 764, *774*
Hazlett, H., 683, *699*
Healey, C. G., 522, *542*
Healey, J., 85, *91*, 439, *456*
Healy, A. F., 53, *59*
Hearst, M. A., 172, *176*, 502, *506*, 567, *572*
Heath, C., 549, 554, *556*
Heath, I., 565, *571*
Heathcote, A., 56, *59*
Heck, H., 864, *867*
Hecker, C., 742, *759*
Heckerman, D., 172, *174*, 187, 191, *197*, 453, *456*
Heckmann, D., 451, *456*
Hedricks, J., 669, *677*
Heeren, E., 605, *618*
Heffley, E., 1227, *1239*
Heffner, T. S., 72, *75*

Hegarty, M., 403, 408, *410*, *411*
Hegedus, E., 706, *716*
Heggestad, E. D., 122, *130*
Heglin, H. J., 723, *738*
Hegner, S. J., 437*n*, 456
Heider, F., 770, *774*
Heim, M., 628, *636*
Heiman, H., 667, *678*
Heirich, A., 626, *636*
Heitmeyer, C. L., 106, *111*
Helander, M., 218, *226*, 574, *598*
Helgeson, V., 548, 549, 556
Heller, R. S., 394, 400, *410*
Heller, W., 85, *90*
Hellerstein, J. L., 493, 503, *505*
Hellman, M., 1015, *1016*
Hellriegel, D., 674, *678*
Helm, R., 1033, *1038*
Helmick, J., 633, *637*
Helsen, W. F., 33, *39*
Helwig, C. C., 1246, *1263*, *1266*
Henderson, A., 170, *172*, 191, *196*, 1066, *1079*
Henderson, V., 890, *893*
Hendrick, H., 323, *325*
Hendrickson, A., 1251, *1263*
Hendrix, A., 356, *377*
Hennecke, M. E., 414, 415, 420, 426, *432*
Hennessy, S., 138, 139, *145*
Henning, R. A., 57, 58, *61*, 647, *656*
Henri, F., 608, 610, *618*
Henrikson, J. O., 139, *144*
Henry, D. L., 84, *90*
Henry, S. L., 1135, *1147*
Henshall, D., 1063, *1080*
Heppner, F., 217, 221, *226*
Hepsø, V., 1065, *1079*
Herbert, L. B., 1140, *1149*
Herbsleb, J. D., 73, *75*, 552, 553, *555*, *556*, *557*, 643, *656*, 1253, *1263*
Heritage, J., 982, *987*
Herlocker, J. L., 442, *457*, 553, *556*, 644, *656*
Herman, J., 1277, *1277*
Hernan, S. V., 547, *556*
Hernandez, A., 475, *486*
Hernandez, M., 780, *792*
Hernandez-Rebollar, J. L., 890, *893*
Herrarte, E., 270, *294*
Herring, S. C., 606, *618*
Herrnson, P. S., 1245, *1261*
Hersey, M. D., 722, *738*
Hersh, H., 924, *926*
Hersh, W. R., 668, *676*
Hershey, J., 180, *199*
Herskovits, M. J., 1252, *1263*
Herskovitz, J., 365, *377*
Hert, C. A., 1248, 1254, *1263*, *1264*
Hertel, G., 583, *598*
Hertzfeld, A., 2, *23*
Hertzog, C., 779, 785, 786, *790*
Hertzum, M., 1138, 1141, *1147*, *1148*, 1177, *1188*
Herz, J. C., 742, *759*
Hess, D., 1178, *1189*
Hess, E. H., 1223, *1238*
Hester, R., 576, *598*
Hetfield, M., 298, *312*
Hettinger, L. J., 415, *429*, 624, 628, *636*
Heuer, F., 81, *90*
Hewison, J., 668, 669, *676*
Hewitt, J., 846, *850*
Hewson, P. W., 139, *146*
Hibbard, J. H., 673, *676*
Hick, W. E., 29, 38, *39*, 48, *59*
Hickey, D. T., 803, 804, *807*
Hickinbotham, K. A., 624, *637*
Hicks, J. E., 832, *850*
Higgins, J., 1226, *1238*
Highsmith, J., 912, 913, *915*
Hightower, J., 179, *197*, 1255, *1263*
Hilbert, D. M., 171, *176*, 442, *455*
Hild, H., 426, *429*
Hildebrandt, N., 496, *506*
Hilgard, E. R., 44, *59*
Hill, C., 1019, *1038*
Hill, E., 68, *76*
Hill, G., 565, *571*
Hill, R., 163, *173*, 1086, *1105*
Hillgren, P. A., 1071, *1077*
Hillman, C. H., 1234, *1238*
Hiltzik, M. A., 2, 7, *23*, 1212, *1219*
Hinckley, K., 162, 164*f*, 165, 166, 167, 168, 170, 171, *172*, *174*, *175*, 180, 181, 185, 193, *197*, 318, *325*, 1110*f*, *1124*
Hind, C. R., 69, *76*
Hindmarch, M., 1172, 1178, 1181, 1182, 1183, 1184, 1185, 1186, 1187, *1188*, *1190*
Hindmarsh, J., 549, 554, *556*
Hinds, P. J., 583, *598*
Hindus, D., 277, *293*, 552, *554*
Hines, J., 473, *486*
Hinterberger, T., 841, *849*, *851*
Hinton, G., 186, *197*
Hintzman, D. L., 53, *59*
Hirano, M., 475, *486*
Hirayama, M., 415, *432*
Hirose, M., 231, *244*, 318, *325*
Hirota, K., 231, *244*, 318, *325*
Hirst, A., 54, *60*
Hirt, E. R., 81, *90*, *91*
Hitch, G. J., 52, *59*
Hix, D., 630, *636*, 1086, *1105*
Hjelm, I., 695, *699*
Hjelm, S. I., 183, *197*
Hludik, F. C. J., 841, *851*
Ho, J., 439, *456*
Hoang, T., 736, *739*, *740*
Hoare, C. A. R., 1085, *1105*
Hoc, J.-M., 574, *598*
Hochberg, J., 409, *410*, 1195, *1207*
Hochheiser, H., 644, *656*
Hockey, G. R. J., 31, 33, *40*, 116, 119, 119*f*, 120, 127, *131*
Hodge, M. H., 723, *738*
Hodges, L. F., 625, 634, *635*, *636*, *638*
Hodgins, J., 474, *485*
Hoedemaeker, M., 706, *717*
Hoel, P. G., 183, *197*
Hofer, E. C., 553, *557*
Hoffman, C., 663, *676*
Hoffman, E. R., 29, *39*
Hoffman, H. G., 139, *145*
Hoffman, R. R., 44, *59*, 121, *130*, 575, 577, 581, *597*

Hoffmeister, R., 887, *893*
Hofstede, G., 365, 366, *378*, 1143, *1148*
Holbrook, C., 1054, *1059*
Holbrook, H., 1054, *1058*
Holdaway, D., 794, *807*
Holden, K., 51, *59*
Holden, M., 634, *636*
Hollan, J. D., 14, *23*, 394, *410*, 795, *807*, 1032, *1038*, 1153, *1169*
Holland, J. H., 56, *59*, 66, *75*
Holland, J. J., 490, 498, *506*
Holland, S., 742, *759*
Hollander, B., 138, *146*
Hollands, J. G., 220, 222, *226*, 449, *458*, 632, *638*
Hollerbach, J. M., 624, 625, *636*
Holley, P., 784, 787, 788, *790*
Hollinger, M., 913, 914, *915*
Hollingshead, A. B., 547, 548, 550, *556*
Hollnagel, E., 150, 153, *157*, 574, 575, 583, *598*, *599*
Holloway, M. S., 1069, *1081*
Holmback, H., 415, *429*
Holmer, T., 1112, *1125*
Holmes, R., 223, *226*
Holmes-Rovner, M., 669, *677*
Holmquist, L. E., 474, *486*, 1036, *1038*
Holmström, J., 1062, 1065, *1079*
Holstein, J. A., 971, *987*
Holtzblatt, K., 729, *738*, 907, 908, 909, 909*t*, *915*, *916*, 922, *926*, 934, 936, *946*, *947*, 950, 951, *963*, 978, *986*, 1047, *1057*, 1062, 1063, 1068, 1073, *1077*
Holyoak, K. J., 56, *59*, 66, *75*
Holzel, T., 204, *226*
Holzman, T. G., 415*n*, 418, *430*, 861, *869*
Homer, A., 280, *292*
Homes-Rovner, M., 669, *677*
Homier, M., 913, 914, *915*
Hommel, B., 50, *59*
Hon, H.-W., 308, *312*, 863, *868*
Honan, M., 168, *174*
Honda, M., 1138, *1149*
Hong, J. I., 192, *196*, 644, *656*, 1033, 1036, 1037, *1038*, *1039*, 1137, *1148*, 1253, *1263*
Hong, L., 417, *430*
Honold, P., 360, *378*
Hood, J. N., 674, *678*
Höök, K., 446, 451*n*, *456*, 1254, *1262*, *1263*
Hoover, A., 1226, *1238*
Hoover, H. J., 633, *636*
Hopper, A., 172, *176*, 180, *198*, 643, *658*
Hopper, G., 3, *23*
Hoppin, S., 780, *790*
Horgan, T., 1072, *1079*
Horgen, G., 218, *225*
Horn, B. K. P., 179, *197*
Horn, D. A., 169, *174*, 190, *197*, 384, *391*
Horn, J. L., 814, *827*
Horn, L. J., 832, *850*
Hornbæk, K., 1138, *1147*, 1185, 1186, *1188*
Hornof, A., 105, 110, *111*, 844, *850*, 1205, *1207*
Horrey, W. J., 705, *717*
Horsky, J., 667, *678*
Horst, H., 975, *987*
Horstmann, T., 642, *656*
Horton, F., 580, 583, *599*
Horton, M., 797, *808*
Horton, T. E., 110, *112*
Horvitz, E., 165, 169, 171, 172, *174*, 179, 180, 182, 187, 190, 191, 192, *196*, *197*, 427, *430*, 436, 439, 442, 453, *456*, *457*
Horwitz, P., 803, *807*
Hoselton, R., 844, *850*
Hospers, H. J., 138, *146*
Houben, G.-J., 864, *867*
Houde, S., 1019, *1038*, *1081*
Houghton, G., 33, *41*
Houghton, R., 426, *432*
Hourcade, J. P., 795, *807*
Hourihan, M., 993, *1016*
Houston, J., 1242, *1263*
Hovel, D., 169, 172, *174*, 187, 191, *197*, 453, *456*
Hovy, E., 399, *410*
Howard, J., 1255, *1263*
Howard, L. A., 33, 34, 35, *40*, *41*
Howard, M., 548, *555*
Howard, S., 274, *293*, 1067, 1068, 1071, *1079*, *1080*
Howarth, P., 219, *226*
Howe, D. C., 653, *656*, 1244, 1251, 1254, 1255, *1262*
Howe, R .D., 233, *244*
Howe, S. R., 45, *61*
Howell, R., 280, *292*
Howells, H., 1233, *1239*
Howells, K., 1068, *1080*
Howes, A., *59*, 68, *76*, 101, 105, *111*, *112*, 1198, *1207*
Hoysniemi, J., 800, 801, *807*
Hsia, P., 1054, *1058*
Hsu, D., 1037, *1038*
Hu, H. W., 606, *619*
Hu, N., 634, *635*
Hu, P. S., 706, *717*, 858, *867*
Huang, J., 179, *197*
Huang, T. S., 179, *197*, 181, 188, *199*, 415, 415*n*, 417, 423, *429*, *431*, *431*, *432*
Huang, W., 549, *556*
Huang, X. D., 426, 427, *428*, *430*, 625, *636*, 862, 863, *868*
Huber, P. J., 184, *197*
Huber, R., 47, *61*
Huberman, A. M., 940, *947*
Hudson, J. M., 1255, *1263*
Hudson, S. E., 439, *456*, 814, *828*, 1033, *1038*, 1123, *1125*, 1249, 1259, *1262*, *1263*
Huenerfauth, M., 873, *882*, 889, *893*
Huettel, S. A., 47, *59*
Huey, B. M., 1195, *1207*
Huff, C., 767, 768, *774*
Huggins, J., 128, *131*
Hughes, D., 283, *293*
Hughes, H. C., 38, *40*
Hughes, J. A., 729, 733, *738*, *915*, 966, 967, 982, *986*, *987*, 1255, *1263*
Hughes, T. P., 1243, *1263*
Huguenard, B. R., 99, *111*
Hulkko, S., 1070, *1079*
Hulstijn, J., 875, *883*
Hultcrantz, J., 1067, *1079*
Hultgren, G., 314, *325*
Humburg, J., 1130, *1149*, 1172, *1189*
Hummel, V., 179, *197*
Humphrey, D. G., 1223, *1238*
Humphreys, B. L., 666, *677*
Humphreys, G., 218, *225*
Humphreys, M. S., 128, *131*

Humphries, T., 886, *893*
Hundt, A. S., 672, *676*
Hung, G., 219, *226*
Hunter, D., 714, *718*
Hunter, M., 168, *176*
Hünting, W., 316, 321, *325*
Hürst, W., 417, *430*
Hutchins, E. L., 14, *23*, 394, *410*, 574, 576, 583, *599*, 982, *987*
Hutchinson, H., 797, *807*
Hux, K., 845, *851*
Hvannberg, E. T., 1185, *1189*
Hwa, H., 172, *176*
Hwang, F., 839, *850*
Hwang, M.-Y., 308, *312*, 840, *868*
Hyde, M., 886, *893*
Hyde, T. S., 53, *59*
Hyder, A. A., 703, *718*
Hyman, P., 744, 745, *759*
Hyman, R., 29, 38, *40*, 48, *59*
Hypponen, H., 823, *827*

I

Iachello, G., 1253, 1255, *1263*
Iacono, S., 1248, *1263*
Iacucci, C., 1071, *1079*
Iacucci, G., 1071, *1079*
Iani, C., 36, *40*
Iavecchia, H. P., 1196, *1207*
Ibrahim, A., 1067, *1079*
Ichicawaa, Y., 549, *557*
Ihde, S., 181, *199*, 414, *432*
Ikar, D., 403, *411*
Illich, I., 122, 123, 129, *131*
Imada, A. S., 1062, 1063, *1080*
Imamoto, K., 182, *197*
Inkpen, K. M., 795, 797, *807*, 1253, *1265*
Inkson, K., 365, *378*
Inselberg, A., 531, *542*, *543*
Intille, S. S., 180, 182, *196*, *198*, 439, *456*
Iqbal, S. T., 439, *456*
Irby, C. H., 472, *486*
Irestig, M., 1069, *1079*
Ironsmith, M., 973, *987*
Irvine, C., 647, *656*, *658*
Isaacs, E. A., 548, 549, 552, *556*, *557*, 1212, *1219*, 1249, 1255, *1263*
Isaacson, D. R., 735, *739*
Isbell, L., 81, *89*
Iscoe, N., 73, *75*
Isdale, J., 628, *636*
Isen, A. M., 81, 82, *90*
Isenhour, P. L., 583, *597*
Isensee, S., 908, *915*, 1135, *1149*
Ishida, S., 243, *245*
Ishii, H., 170, 171, 174, 180, 194, *196*, *197*, *198*, 284, *294*, 470, 473, 474, 475, 478, 480, 481, 484, 485, *486*, *487*, 795, *808*
Isomursu, M., 1070, 1071, *1079*
Israelski, E., 9, *23*
Istance, H., 219, *226*, 843, *849*
Ito, M., 1245, *1264*
Ito, T. A., 85, *89*
Itoh, K., 846, *850*
Itoh, Y., 474, *486*
Iversen, O. S., 797, 799, *807*, *808*, 1072, *1079*
Ivory, M. Y., 502, *506*
Iwasaki, S., 725, *739*
Iwasaki, T., 218, *226*
Iwase, H., 784, *791*
Iwata, H., 230, 231, 233, 238, 240, *244*
Iyengar, G., 414, *430*
Izadi, S., 277, *294*
Izard, C. E., 79, 87, *90*

J

Jacko, J. A., 52, *59*, 310, *311*, *312*, 384, *391*, 675, *677*, 835, *851*, 854, 854*f*, 859, 861, *867*, *868*, *869*, *870*, 1254, *1263*
Jacks, H., 80, *89*
Jackson, A., 733, *740*
Jackson, B. M., 547, *555*
Jackson, C., 773, *774*
Jackson, J., 595, *597*
Jackson, S. L., 804, *808*
Jackson, S. R., 34, *41*
Jacob, R. J. K., 38, *40*, 168, *174*, 181, *197*, 474, 475, , *486*, *487*, 842, *852*
Jacobs, A., 169, *174*, 1251, *1261*
Jacobs, J., 574, *599*
Jacobs, K. W., 84, 87, *90*
Jacobsen, N. E., 1138, 1141, *1148*, 1177, 1181, *1188*
Jacobson, I., 993, *1016*, 1055, *1058*
Jacobson, L., 766, 769, *774*
Jacoby, L. L., 53, *60*
Jadad, A. R., 670, *677*
Jaervinen, P., 1063, *1081*
Jagacinski, R. J., 576, 581, *599*
Jagadeesan, L. J., 552, *555*, 1253, *1263*
Jahanian, F., 552, 557
Jain, A. K., 182, 193, *197*, *198*, 414, 417, 428, *430*, *431*
Jalili, R., 426, 429
James, C. A., 315, *324*
James, J., 934, 935, 943, *947*
James, W., 85, *90*
Jameson, A., 443, 449, *456*
Jamieson, B. A., 787, *791*
Jancke, G., 172, *173*, 549, *555*, 1253, *1263*
Janecek, P., 1030, *1039*
Janelle, C. M., 1234, *1238*
Janis, F. T., 1212, 1215, *1219*
Janis, I., 579, *599*
Jannson, J., 204, *225*
Jansen, B. J., 565, 566, *572*
Janssen, T., 521, *542*
Janssen, W., 550, *555*
Janssen, W. H., *555*, 706, *717*
Jarawan, E., 703, *718*
Jarke, M., 1046, 1057, *1058*, *1059*
Jarvenpaa, S. L., 673, *677*
Jaschinski, W., 218, 219, *226*
Javitt, J. C., 858, *867*, *868*, *869*
Jaworski, B. J., 138, 146
Jay, G. M., 785, *791*
Jeannerod, M., 29, *40*
Jebara, T., 417, *429*
Jeffries, R., 1178, 1179, 1181, *1188*
Jellinek, H., 167, *174*
Jenkins, C. R., 1153, *1168*

Jenkins, H., 743, *758*
Jenkins, J. J., 53, *59*
Jenkins, L., 872, *882*
Jennings, K. R., 1077, *1078*
Jensen, C., 553, *556*, 644, *656*, 799, *807*
Jensen, F. V., 187, *197*
Jensen, J. J., 794, *806*, *808*
Jentsch, F., 574, *597*
Jëonsson, A., 420, *429*
Jessome, D. R., 180, *198*
Jessup, L., 15, *23*
Jewett, J. J., 673, *676*
Ji, Q., 439, *456*
Jiang, L., 862, *868*
Jiang, X., 1253, *1263*
Jimison, H. B., 138, *145*
Jiwnani, K., 876, *882*
Johansen, R., 547, *557*
Johanson, A., 846, *850*
Johansson, E., 708, *717*
Johansson, G., 182, *197*
Johansson, M., 1073, *1079*
John, B. E., 12, *23*, 95, 98, 100, 101, 108, *111*, *112*, 166, *174*, 286, *293*, 490, 498, *505*, *506*, 929, *947*, 1053, *1058*, 1138, 1140, 1141, *1147*, *1148*, 1177, 1178, 1181, 1184, *1188*, 1195, 1198, 1199, 1201, 1202, 1203, 1204, 1205, *1206*, *1207*
John, D. R., 674, *677*
Johnson, B., 1135, *1147*
Johnson, C., 274, 275, *293*
Johnson, D. G., 1033, 1043, *1059*, 1247, 1248, 1249, 1257, 1259, *1261*, *1263*
Johnson, E. J., 673, *677*
Johnson, E. N., 734, *739*
Johnson, G., 644, 649, *656*
Johnson, H., 547, 563, *572*, 1087, *1105*, 1184, *1188*
Johnson, J. C., 31, *39*, 472, *486*, 576, *656*, 970*n*, 973, *987*
Johnson, M. H., 495, *506*
Johnson, M. M. S., 669, *677*
Johnson, O., 87, *92*
Johnson, P., 496, *599*, 1087, *1105*
Johnson, R., 1033, *1038*
Johnson, T. J., 13, *23*, 142, *145*
Johnson, W., 879, *882*
Johnson-Laird, P. N., 66, 67, *75*, 79, *91*, 731, *739*
Johnson-West, 875, *882*
Johnston, J., 121, *132*
Johnston, J. C., 576, *600*
Johnston, J. H., 121, *130*, *131*
Johnston, M., 415, 427, *428*, *429*, *430*
Johnston, W. A., 576, *599*
Joiner, R., 768, 769, *774*, 795, *808*
Joines, S., 218, *227*
Jojic, N., 170, *174*, 188, *197*
Jolicœur, P., 56, *61*
Jonassen, D., 575, *599*
Jones G. B., 604, *618*
Jones, A., 180, *198*
Jones, C. M., 87, *90*, 439, *456*
Jones, D. E., 81, *91*
Jones, D. G., 734, *739*
Jones, D. R., 574, *599*
Jones, E. E., 140, *145*, 770, *774*, 578, *599*
Jones, G., 96, *112*
Jones, J., 669, *677*
Jones, L., 499, *506*, 624, *636*
Jones, M. A., 171, *174*, 837, 845, *851*
Jones, M. B., 625, *637*
Jones, P. M., 575, 580, *599*, *601*
Jones, R. E., 724, *738*
Jones, R. M., 100, 101, 110, *112*
Jones, S., 65, *74*, 606, *618*, 934, *947*, 1199, 1200, *1208*, 1254, *1263*
Jones, S. B., 634, *638*
Jones, S. J., 1026, *1038*
Jones, T., 168, *175*, 796, *808*
Jonides, J., 31, *40*, 1227, *1239*
Jonsson, I., 439, *456*
Jonsson, P., 993, *1016*
Joo, H., 781, *790*
Jordan, B., 548, *557*, 972, *987*
Jordan, M., 187, *197*
Jordan, P. W., 684*t*, 685, 697, *699*, 1145, 1146, *1148*
Jorge, J., 1102, *1105*
Joroff, M. L., 1072, *1079*
Jorve, W. R., 724, *739*
Jøsang, A., 652, *657*
Jose, P. E., 83, *91*
Joseph, M., 1252, *1262*
Josephson, J., 593, *599*
Josephson, S., 593, *599*
Jouault, F., 1121, *1124*
Joy, K., 250, *264*
Joyce, B. J., 788, *791*
Joyce, J. B., 786, *790*
Joyner, L. A., 128, *131*
Juang, B., 181, *198*
Judge, T. A., 127, *130*
Jul, S., 563, *572*
Julia, L., 427, *429*
Jung, Y., 1256, *1265*
Jungeblut, A., 872, *882*
Jungk, R., 1067, *1079*
Junker, B. W., 99, *111*
Junker, H., 182, *196*, 890, *892*
Junqua, J. C., 191, *197*
Just, M., 650, *657*
Just, M. A., 99, *112*, 403, 408, *410*, 1223, *1238*
Juth, S., 427, *432*

K

Kaarstad, M., 153, *157*
Kabbash, P., 166, 169, 170, *174*
Kaczala, C. M., 771, *774*
Kadie, C. M., 187, *197*
Kadous, M. W., 890, *893*
Kahn, D., 646, 653, *657*
Kahn, H., 1057, *1058*
Kahn, J. M., 179, *197*
Kahn, M. G., 924, *926*, *946*
Kahn, P. H., Jr., 83, *90*, 646, *656*, 1245, 1247, 1250, 1251, 1252, 1253, 1254, 1256, 1255, *1261*, *1262*, *1263*, *1264*
Kahneman, D., 55, 57, *60*, 124, *131*, 578, 595, *598*, *599*, *602*, 1043, *1058*, *1059*
Kaikkaner, A., 1138, *1148*
Kail, R., 794, *808*
Kaimullah, A., 384, *391*
Kaindl, H., 1054, 1055, *1058*, 1065, *1079*

Kaiser, E., 428, *428*
Kajimoto, H., 233, *244*
Kakay, R. A. E., 842, 849, *851*
Kalasky, M. A., 785, 788, *791*
Kalawsky, R. S., 625, 626, *636*
Kalliot, T., 1138, *1148*
Kallman, E. A., 630, *636*, 643, *657*
Kaluger, G., 813, 816, *827*
Kamachi, M., 85, *90*
Kamba, T., 171, *174*
Kameenui, E., 877, *882*
Kamm, C., 387, *391*, 549, *556*
Kammerer, E. E., 73, *75*
Kammersgaard, J., 1065, *1078*
Kanade, T., 85, *92*
Kancherla, A., 630, *637*
Kandogan, E., 648, *657*
Kaneko, F., 243, *245*
Kanevsky, D., 891, *892*
Kanifer, A., 139, *146*
Kankainen, A., 1138, *1148*
Kant, I., 1245, *1264*
Kantner, L., 974, *987*
Kantowitz, B. H., 711*t*, *716*, 724, *738*
Kao, E., 13, *23*
Kaplan, C., 569, *572*
Kapoor, A., 554, *557*
Kappelman, L., 1077, *1079*
Karagiannidis, C., 864, *870*
Karahanna, E., 901, *915*
Karande, S., 874, *882*
Karasti, H., 971, *987*, *1079*
Karat, C., 169, *174*, 384, *390*, *391*, 640, 642, 643, 644, 645, 652, 653, *655*, *656*, *657*, 1180, *1188*, 1268, 1270, *1277*
Karat, C.-M., 190, *196*, *197*, 318, *325*, 835, 838, 844, 847, *851*, *852*
Karat, J., 169, *174*, 190, *197*, 384, *391*, 640, 642, 643, 644, 645, 652, 653, *655*, *657*, 835, 838, 844, 847, 862, *867*, 921, 924, *926*, 1046, 1047, *1057*, *1058*, 1065, *1079*
Karimullah, A., 835, 838, 844, 847, *850*, *852*
Karis, D., 1052, *1059*, 1135, *1149*
Kärkkäinen, L., 53, *60*
Karl, L., 169, *174*
Karpouzis, K., 889, *893*
Karsenty, L., 549, *556*
Karsh, B., 315, *324*, 667, *677*
Karshmer, A. I., 862, *868*
Karttunen, F., 1064, *1079*
Karwowski, W., 46, *59*, 702, *718*
Karyukin, B., 1141, *1148*
Kasabach, C., 298, *311*
Kasippillai, S., 280, *292*
Kass, R. E., 99, *111*
Kass, S. J., 1226, *1238*
Kassel, N., 180, *197*
Kassell, N. F., 162, 167, *174*
Kasser, T., 122, *131*
Kassin, S. M., 576, *597*
Katcher, A., 1255, *1261*
Kato, H., 182, *197*, 474, *487*
Katriel, A., 501, *506*
Katz, E., 88, *90*
Katz, N., 583, *599*
Katz, R. H., 179, *197*
Kaufman, A. S., 813, *827*
Kaufman, D. R., 674, *677*
Kaufman, L., 206*f*, *226*
Kaufmann, R. J., 6, *22*
Kaugars, K., 862, *868*
Kauppinen, M., 943, *947*
Kaur, K., 631, *637*
Kaushal, R., 662, *677*
Kawahara, T., 439, 440*f*, *456*
Kawai, S., 863, *868*
Kawai, Y., 233, *244*
Kawakami, N., 233, *244*
Kawato, S., 843, *850*
Kay, A., 7, *23*, 137, *145*, 797, *808*
Kay, J., 435, *456*, *457*
Kaye, B. K., 142, *145*
Kaye, H. S., 783, *791*
Kaye, J., 1251, *1265*
Kayl, K., 702, *718*
Kazman, R., 863, *869*, 1121, *1124*
Ke, X., 650, *657*
Keahey, T. A., 538, *543*
Keates, S., 837, 839, 840, 847, *850*
Keeker, K., 680*t*, *700*, 1141, 1145, *1148*
Keele, S. W., 28, *40*
Keen, P. G. W., 6, *23*
Keim, D. A., 526, *543*
Keinath, A., 710, *718*
Keinonen, T., 1070, *1079*
Keirn, Z. A., 841, *851*
Kekalainen, A., 1138, *1148*
Keller, A., 493, 503, *505*
Keller, D. S., 1021*f*, *1039*
Kellerman, H., 87, *91*
Kellert, S. R., 1255, *1263*
Kelley, H. H., 771, *774*
Kelley, J. F., 1026, *1038*
Kelley, T., 912, *915*
Kellogg, W. A., 1046, 1049, 1054, *1057*, *1059*, 1255, *1263*
Kelly, C. L., 785, 786, *790*, *791*
Kelly, D., 441*n*, *456*
Kelly, R. V., 608, *618*
Kelsey, J., 646, *657*
Kelso, J. A. S., 29, *40*
Kemmerer, R., 647, *657*
Kemp, J. A. M., 801, *808*
Kendon, A., 415, 420, 423, *430*, 549, *556*
Kenkeremath, D. C., 1212, 1215, *1219*
Kennaway, R., 889, *893*
Kennedy, A., 219, *226*
Kennedy, J. M., 665, *677*, 1167, *1168*
Kennedy, K. E., 625, 627, 628, 629, 630, *637*
Kennedy, P. R., 842, 849, *851*
Kennedy, R. J., 152, *157*
Kennedy, R. S., *637*, *638*
Kennedy, S., 1136, *1148*
Kenny, P., 96, *112*
Kensing, F., 575, *597*, 982, *987*, 1062, 1063, 1065, 1067, 1068, 1074, *1078*, *1079*, 1250, *1264*
Kent, S. L., 742, *759*
Keough, K. A., 770, *773*, 1268, *1277*
Kephart, J. O., 435, *457*
Keppel, G., 52, *60*
Kerick, S. E., 1234, *1238*
Kern A., 188, *196*

Kernal, H. K., 83, *91*, 138, *145*
Kerr, B., 435*n*, *456*
Kerr, R., 71, *75*
Kershner, J., 626, *637*
Kesselman, C., 269, *292*, 547, *555*
Kessler, G. D., 626, *637*, 634, *638*
Kesteren, I. V., 1181, *1189*
Keyser, G. L., Jr., 725, *737*
Khan, A., 171, *173*, *174*
Khaslavsky, J., 134, *145*
Kickbusch, I., 873, *882*
Kiekel, P. A., 58, *60*
Kieras, D. E., 47, 56, 58, *60*, 66, *75*, 95, 98, 99, 104, 104*f*, 105, 106, 107, 110, *111*, *112*, 166, *174*, 502, 503, *505*, 575, *599*, 1053, *1058*, 1195, 1198, 1199, 1201, 1202, 1203, 1204, 1205, 1206, *1207*
Kiern, Z. A., 183, *197*
Kiernan, M., 1167, *1168*
Kiernan, N. E., 1167, *1168*
Kiesler, S., 139, *146*, 439, *457*, 547, 548, 551, 553, *555*, *556*, *557*, 583, *598*, 1242, 1248, 1249, 1255, *1262*, *1264*, *1266*
Kilander, B., 1015, *1016*
Kilander, F., 262, *264*
Kilanski, K., 181, *196*
Kilduff, M., 972, *987*
Kilgore, R., 142, *145*
Kilgourz, A., 267, *293*
Killen, M., 1246, *1266*
Kim, A. J., 563, 684*t*, 699
Kim, L., 318, *324*
Kim, N., 143, *145*
Kimmel, S. E., 667, *677*
Kindfield, A. C. H., 803, *807*
Kindlund, E., 1138, *1148*, *1149*
King, C. N., 214, 215*f*, *226*, 889, *893*
King, J. L., 1248, *1264*
King, P., 134, *145*
King, T. W., 837, 846, 848, *851*
Kingdon, K., 629, *638*
Kinshuk, *568*, 572
Kintsch, A., 847, 848, *851*, 1077, *1078*
Kintsch, E., 67, *75*
Kintsch, W., 71, *75*, *76*, 102, *112*
Kirakowski, J., 1138, *1148*
Kirby, K. C., 139, *145*
Kirchner, C., 854, *868*
Kirchoff, K., 185, *196*
Kirikci, O., 842, *850*
Kirkpatrick, A., 166, *173*
Kirkup, L., 841, *850*, *851*
Kirlik, A., 575, *597*
Kirsch, I., 872, 874, *882*
Kirschenbaum, S. S., 94, *111*
Kirsh, D., 170, *174*
Kirwan, B., 152, 153, *157*, *158*, 575, *599*, 929, *947*, 1199, 1202, 1205, 1206, *1207*
Kishino, F., 474, *486*
Kissel, G. V., 1276, *1277*
Kistler, J. J., 287, *293*
Kitajima, M., 102, 103, 111, *112*
Kitamura, Y., 474, *486*
Kitchin, R., 283, *292*
Kittowski, F. F., 607, *618*
Klær, A., 1072, *1079*
Klatzky, R. L., 244, *244*, *245*
Klawe, M., 797, *807*, *882*, 1077, *1080*
Klayman, J., 579, *599*
Klein, D. J., 55, 82, 85, *89*
Klein, G. A., 575, 577, 579, 581, *597*, *599*
Klein, J., *90*, 178, *197*, *808*
Klein, J. T., 1076, *1079*
Klein, M. L., 547, *555*
Klein, R. A., 139, *145*
Klein, R. M., 33, *41*, *60*
Kleinberg, J. M., 566, 567, *572*
Kleinginna, A. M., 78, *90*
Kleinginna, P. R., Jr., 78, *90*
Kleinlein, J., 211, *227*
Kleinmuntz, D. N., 580, *599*, 673, *678*
Klemmer, S. R., 475, *486*, 1037, *1038*
Klima, E. S., 886, *893*
Kline, D. W., 784, *791*, 858, *868*
Kline, R. L., 865, *868*
Kiesler, S., 139, 140, *146*
Kling, R., 179, *197*, 1248, 1249, *1263*, *1264*
Klinger, A., 491, *506*
Klinker, G., 625, *635*
Klopfer, D., 580, *599*
Knapp, B. R., 841, *851*
Knapp, R. B., 837, 841, *851*
Knave, B., 314, *324*, *325*
Knerr, B. W., 625, *637*
Knight, 365, *564*
Knight, S., 379, *572*
Knipling, R. R., 704*t*, *719*
Knoll, R. L., 50, *61*
Knorr, D., 735, *739*
Knox, H., 975, *986*
Knox, S., 934, *947*
Knudsen, J. L., 1065, *1078*, 1025, *1038*
Ko, A. J., 439, *456*
Kobayashi, K., 475, 479, *486*
Kobayashi, M., 565, *572*
Kober, H., 168, *175*
Kobrin, S., 138, *146*
Kobsa, A., 414, *430*, 439, 441*n*, 451, *455*, *456*, 642, 643, *657*, *658*, 845, *851*, 1090, *1105*
Kobus, D. A., 1223, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1235, *1238*
Koch, C., 725, *740*
Koch, P., 439, *456*
Kochhar, D., 708, *718*
Kochoubey, B., 841, 849, *851*
Koedinger, K. R., 108, *111*, 445, *455*, 1201, *1207*
Koehler, J. J., 578, *599*
Koelega, H. S., 127, *131*
Koenemann, J., 566, *572*
Koerner, B., 475, *486*
Koester, H. H., 837, 839, 846, 847, *851*
Koestner, R., 123, *130*
Kogan, S., 549, *556*
Koh, T. K., 171, *176*
Köhler, A., 607, *618*
Kohler, T., 428, *429*
Kohler, W., 576, *599*
Kohls, L. R., 365, *379*
Kohno, M., 171, *175*
Kokoschka, S., 220, *226*
Kolatch, E., 878, *882*
Kolbert, E., 607, *618*

Koleva, B., 554, *555*
Kolli, R., 1065, *1081*
Kollias, S., 85, 86, *89*
Kollmorgen, G. S., 300, *311*, 1222, 1223, *1238*
Kollmorgen, L. S., 1222, *1238*
Kollock, P., 553, *556*
Kolm, P., 1063, *1078*
Kolodner, J., 579, 581, *599*, 797, 806, *808*
Kolojejchick, J. A., 536, *542*
Kolson, C. L., 813, 816, *827*
Kolstad, A., 872, *882*
Kolvenbach, S., 551, *557*
Komatani, K., 439, 440*f*, *456*
Komlodi, A., 651, *656*
Kompfner, P., 710, *718*
Kondziela, J. M., 1141, *1147*, 1179, *1188*
König, W., 706, *718*
Konig, Y., 417, *429*
Konold, C., 502, *506*
Konomi, S., 1112, *1125*
Konrad, T. R., 674, *677*
Konradt, U., 583, *598*
Konstan, J. A., 448, *458*, 553, *556*, 642, *656*
Kontarinis, D. A., 233, *244*
Kooijam, A. C., 858, *867*
Koons, D., 415, 417, *430*
Kooper, R., 192, *196*
Kopardekar, P., 735, *739*
Kopp, S., 415, *430*
Koppel, R., 667, *677*
Korn, P., 1254, *1261*
Kornblum, S., 29, *40*, 49, 50, *60*
Korpela, M., 617, *619*, 1250, *1264*
Kortuem, G., 191, *196*
Korzenny, F., 604, *619*
Kosbie, D. S., 498, *506*, 1035*f*, *1039*
Koschmann, T., 553, *556*, 805, *808*
Koseki, Y., 182, *198*
Koskinen, I., 977, *987*
Koss, F. V., 96, *112*
Koss, S., 665, 670, *677*
Kosslyn, S. M., 522, *543*
Kostrzewski, A., 204, *225*
Kostyniuk, L. P., 704, *716*
Kothiyal, K., 218, *226*
Kotovsky, K., 582, *599*
Koumpis, A., 864, *870*
Koushki, P. A., 705, *718*
Kovacs, G. T., 180, *197*
Kovath, K., 873, 876, *883*
Koved, L., 426, *429*
Krabbel, A., 1066, *1079*
Kracher, B., 652, *656*
Kraemer, H. C., 1166, *1169*
Kraemer, K. L., 546, 550, *556*
Krahmer, E., 1140, *1148*
Krajcik, J., 879, *882*
Kramer, A. F., 1223, 1227, *1238*, *1239*
Kramer, G., 248, 249, 261, *264*
Kramer, K., 465, *487*
Krampe, R. T., 64, *75*
Krapichler, C., 675, *677*
Krasner, E. G., 1033, *1038*
Krasner, H., 73, *75*
Krause, A., 311, *311*
Krauss, R. M., 548, *556*
Kraut, R. E., 73, 75, *75*, 548, 549, 552, *555*, *556*, 1249, *1262*
Kray, C., 449, *455*
Krebs, M. J., 49, *61*
Krebs, V., 610, *619*
Krekelberg, A., 644, *656*, 1253, *1263*
Kreuter, M., 138, *146*
Krichever, M., 298, *312*
Kricos, P. B., 423, *430*
Kriegel, H.-P., 526, *543*
Krishnan, M .S., 551, *557*
Kristensson, P., 168, *174*
Kristof, R., 394, 404, *411*
Kroll, K., 742, *759*
Kropf, S., 219, *226*
Krosnick, J. A., 1155, *1169*
Krovetz, R., 564, *572*
Krueger, H., 222, *226*, 475, *487*
Krueger, R. A., 757, *759*, 910, 911, *915*
Krüger, A., 451, *456*
Kruger, R., 194, *197*
Kruijff, E., 162, *173*
Kruk, R. S., 822, *828*
Krull, H., 277, *293*
Krulwich, B., 449, 449*f*, *456*
Krum, D., 310, *312*
Krumm, J., 172, *174*, 179, 188, 192, 193, *196*, *197*
Kruppa, M., 451, *456*
Kruse, A. A., 300, *311*, 1222, 1223, 1233, *1239*
Kubeck, J. W., 787, *791*
Kübler, A., 841, 849, *849*, *851*
Kucan, J., 289, *293*
Kucera, H., 492, *506*
Kuchar, J. K., 732, *739*
Kuchinsky, A., 549, 550, 552, *556*
Kuehn, L., 667, 668, *676*, *677*
Kuh, G. D., 1167, *1168*
Kuhlen, R., 651, *657*
Kuhn, A., 357, *379*
Kuhn, K., 169, *175*, 419, 420, 425, *431*
Kuhn, S., 1062, 1063, *1080*, 1250, *1264*
Kujala, S., 943, 945*t*, *947*
Kulak, D., 574, *599*
Kulik, C. T., 141, *146*
Kulikowski, C., 427, *432*
Kulkarni, M., 874, *882*
Kulkarni, Y., 417, *430*
Küller, R., 219, *226*
Kumar, R. R., 564, *571*, 664, 667, *676*, *677*
Kumar, S., 427, *430*, 664, *677*
Kumaraguru, P., 645, *657*
Kung, D., 1054, *1058*
Kuniavsky, M., 909*n*, 914, *915*
Kuo, J., 614, *619*
Kuorelahit, J., 299, *312*
Kuperman, G. J., 667, 668, *676*
Kurimoto, S., 218, *226*
Kurlander, D., 87, *90*
Kurosu, M., 1138, *1149*
Kurtenbach, G., 163, 165, 167, 169, 170, 171, 172, *172*, *173*, *174*, *175*, *176*, 551, *556*, 1032, *1039*
Kurtev, I., 1115, *1125*
Kurz, K., 81, *91*

Kurze, M., 862, *868*
Kusano, M., 795, *808*
Kushleyeva, Y., 582, *599*
Kushniruk, A. W., 674, *677*
Kussmann, H., 710, *718*
Kuutti, K., 1048, 1054, *1058*, 1070, 1071, *1079*
Kuzuoka, H., 549, *556*
Kveraga, K., 38, *40*
Kwok, J. Y.-C., 1070, *1079*
Kyle, B. D., 1035, *1039*
Kylian, H., 219, *226*
Kyng, M., 15, *22*, 1021, 1025, *1038*, 1046, 1048, 1049, *1058*, 1062, 1063, 1065, 1066, 1069, 1071, 1072, 1075, *1077*, *1078*, *1079*, 1250, *1262*, *1263*, 1250, *1264*
Kyriakooulos, N., 890, *893*

L

Laarni, J., 53, *60*
Labaw, P., 754, *759*
Lachman, J. L., 45, *60*
Lachman, R., 44, *60*
Lackner, J. R., 634, *636*, *637*
LaCourse, J. R., 841, *851*
Lafreniére, D., 1069, 1070, 1073, *1079*
Lagoudakis, M., 844, *851*
Lai et al., 563
Lai, F. C. H., 713, *718*
Lai, J., 384, 385, *391*, 563, *572*, 1253, *1265*
Lai, K. Y., 548, *556*
Laike, T., 219, *226*
Laird, J. E., 96, *111*, *112*, 503, *506*, 575, 581, *599*, *600*, 742, *759*
Lallouache, M.-T., 415, *428*
Lallouache, T., 423, *431*
Lalmas, M., 566, *572*
LaMarca, A., 179, *197*, 1253, 1255, *1262*
Lamberton, J., 889, *893*
Lamberty, K. K., 797, *808*
Lamias, M. J., 1167, *1168*
Lamola, S., 970, *987*
Lamont, M., 366, *378*
Lamping, J., 539, *543*
Lan, T., 1228, *1238*
Landau, R., 10, *23*
Landauer, T. K., 70, *75*, 103, *112*, 496, 501, *506*, 574, *598*, 780, 969*n*, *987*, 1181, 1185, *1189*, 1192, 1197, *1207*
Landay, J. A., 165, *175*, 475, *486*, 644, *656*, *869*, 1033, 1035, 1036, 1037, *1038*, *1039*, 1135, 1137, *1148*, *1149*, 1253, *1263*
Landry, S. J., 736, *739*
Lane, H., 886, 887, *892*, *893*
Lane, N. E., 628, *637*, 1196, *1207*
Lang, F., 222, *226*
Lang, P. J., 79, 87, *90*
Langdon, P., 837, 839, 840, 847, *850*
Lange, B. M., 551, *556*
Langlana, L., 230, *244*
Langley, P., 451, *456*
Langston, W. E., 68, 69, *75*
Langton Hewer, R., 833, *852*
Lanier, J., 448, *456*
Lank, E., 166, 169, *175*, *176*
Lankford, C., 846, *851*
Lanning, S., 276, *294*
Lansdale, M., 172, *175*
Lansdale, M. W., 153, *157*
Lanz, H., 202, *225*
Lanzara, G. F., 1065, *1079*
LaPlante, M. P., 856, *868*
Laraki, O., 142, *145*, *279*, *292*, 652, *656*
Laramee, R., 310, *312*
Larkin, J. H., 66, *75*, 517*t*, 580, 582, *599*, 1202, *1207*
Larman, C., 912, 912*t*, 913, *915*
Larrick, R. P., 578, *597*
Larsen, J. T., 85, *89*
Larsen, S., 218, *225*, 876, *882*
Larson, A., 574, 576, 583, 596, *599*
Larson, J., 863, *869*, 878, *882*
Larson, K., 182, *198*, *882*
Larsson, M., 814, *827*
Lassen, M., 1024, *1038*
Lasswell, T. E., 171, *173*
Lathoud, G., 428, *429*, *430*
Latour, B., 968, *987*, 1242, *1264*
Lau, T., 450, *458*
Läubli, T., 316, 321, *325*
Laudauer, T. K., 780, *790*
Laughery, K. R., 1194, *1207*
Lausen, G., 448, *458*
Lave, J., 982, *987*
Lavery, D., 267, *293*, 1131, *1147*, 1173, 1174, 1177, 1178, 1180, 1181, 1182, 1187, *1188*, *1189*
Lavie, N., 54, *60*
Lavie, T., 403, *411*
Lavin, B., 612, *618*
LaViola, J., 162, *173*, 192, *197*
LaViola, R., 169, *175*
Law, C., 1140, *1148*
Law, E. L.-C., 1185, *1189*
Law, S. A., 547, *555*
Lawler, E. E., 323, *325*
Lawrence, D., 12, *23*, 490, *506*
Lawrence, S., 564, *572*
Lawson, B. D., 624, *637*
Lawton, J., 874, *881*
Layton, C., 574, 576, 580, 583, 596, *601*
Lazar, J., 563, *572*, 644, *656*, 1254, *1264*
Lazarus, R. S., 83, *90*, 116, 117, *131*
Lazer, D., 583, *599*
Lazzaro, J. J., 890, *893*
Lazzaro, N., 680, 680*t*, 685, 686, 687, 688*f*, 690*f*, 691*f*, 692, 693, 694, 694*f*, *700*
Le Bihan, D. 1234, *1238*
Le Goff, B., 414, 415, 420, *428*
Leadley, B., 914, 914*t*, *915*
Leape, L. L., 667, 668, *676*
Leatherby, J. H., 419, *430*
Leavitt, J. L., 29, *40*
Lebei, L., 547, *555*
Lebiere, C., 58, *58*, 106, 107, *111*, 575, *597*
LeBlanc, M., 684*t*, *700*
Lederer, S., 644, *656*, *657*
Lederman, S. J., 233, *244*, 244, *245*
LeDoux, J. E., 78, 78*n*, *90*
Leduc, N., 277, *293*
Lee, A., 500, *507*, 1143, *1147*
Lee, B., 1123, *1125*, 1245, 1255, *1261*
Lee, C. C., 415, *430*, 785, *791*, *792*

Lee, D., 550, *555*
Lee, E., 55, *60*
Lee, F., 574, *599*
Lee, F. J., 109, *113*
Lee, F. W., 666, *678*
Lee, H. C., 153, *157*
Lee, H.-W., 876, *868*
Lee, J., 58, *62*, 171, *175*, 475, *486*, 583, 590, *598*, *599*, *602*, 624, 667, 668, *676*, 858, *882*
Lee, J. C., 305, *312*
Lee, J. W., 624, *636*
Lee, K. K., 733, 734, 735, *739*
Lee, K.-F., 305, *312*
Lee, R., 474, *486*, 1065, *1080*
Lee, S., 890, *893*
Lee, S. W., 180, *197*
Lee, T. D., 38, *40*, 70, *75*
Lee, W. O., 941, *947*
Lee, Y. S., 360, *378*
Lee, Y. V., 417, *432*
Legge, G. E., 858, *868*
Legler, J. D., 666, *677*
Legro, M. W., 858, *867*, *868*
Lehericy, S., 1234, *1238*
Lehman, J. F., 101, *112*
Lehner K., 684, *699*
Lehner, P. E., 582, *599*
Leichner, R., 549, *556*
Leigh, D., 171, *173*, *175*, 181, *196*, 474, *485*
Leigh, J., 554, *557*
Lemke, A. C., 593, *598*
Lemmens, P., 258, *264*
Lemos, R., 634, *636*
Lemstrom, K., 491, *507*
Lenat, D., 13, *23*
Lenert, L. A., 668, *678*
Lengnick-Hall, M. L., 1077, *1078*
Lentz, R., 409, *411*
Lenzmeier, S., 764, *774*
Leonard, D., 977, *987*
Leonard, V. K., 859, *868*, *869*
Leonardi, R., 1254, *1265*
Leonhardt, U., 475, *487*
Leont'ev, A. N., 615, *619*
Lepistö, A., 1143, *1148*
Leplat, J., 574, *600*
Leplâtre, G., 258, *264*
Lepper, M. R., 136, *146*, 746, *759*, 767, 772, *774*
Lerch, F. J., 99, *111*, 141, *145*, *146*
Lerch, J. F., 73, *75*
Lerner, R. A., 1123, *1124*
Lesaigle, E. M., 1137, *1148*
Lesgold, A., 580, *599*
Lesh, N., 194, *198*, 437, 438, 448, *456*, *457*
Lesko, J., 1212, *1219*
Lester, J., 182, *197*
Letchner, J., 179, *197*
Leung, Y. K., 538, *543*
Levenson, R. W., 85, *89*, *90*
Leveson, N. G., 1252, *1264*
Levesque, H., 578, 595, *597*
Levesque, L. L., 74, *75*
Levi, D., 576, 579, 583, *599*
Lévi-Strauss, C., 1044, *1058*
Levie, W. H., 409, *411*
Levin, H., 772, *774*
Levin, I. P., 647, *677*
Levin, T., 647, *658*
Levine, K., 747, *759*
Levine, S. P., 837, 839, 847, *851*
Levine, T., 647, *656*
Levinger, D., 1062, *1079*
Levinson, M., 993, *1016*
Levison, W., 711*t*, *717*
Levitt, R. E., 667, *676*
Levow, G. A., 419, *431*, 862, *870*, 389, *391*
Levy, B. I., 84, *90*
Levy, J. L., 732, *739*, 1138, *1148*
Levy, P. S., 1199, 1200, *1208*
Lewenstein, M., 1139, *1148*
Lewis, 780, *792*
Lewis A., 1160, *1168*
Lewis J. R., 204, *227*
Lewis, C. H., 18, *23*, 95, *111*, 101, *112*, *506*, 575, 576, *599*, 702, *716*, 1047, 1053, *1058*, *1059*, 1175, *1189*, 1175, 1176, *1189*, *1190*
Lewis, D., 580, *599*
Lewis, J., 167, 168, *175*, 426, *429*, 1140, *1148*
Lewis, M., 583, *602*
Lewis, R., 365, *379*
Lewis, R. L., 101, *112*, 1198, *1207*
Lewis, S., 29, *39*
Li, D., 439, *456*, 1253, *1263*
Li, F., 552, *557*
Li, I., 1036, *1038*
Li, J., 1037, *1038*
Li, K., 171, 172, *173*, *176*
Li, N., 901, *916*
Li, S. Z., 182, 193, *197*
Li, X., 181, *196*, 439, *456*
Li, Y., 165, 169, 170, *175*, 1036, 1037, *1039*
Liang, T.-P., 574, *602*
Liao, C., 172, *175*
Liberman, M. L., 837, *850*
Licht, B. G., 771, *774*
Lichtenstein, E. H., 887, *893*
Licklider, J. C. R., 7, *23*, 1222, *1238*
Liddy, E. D., 1254, *1263*
Lie, K., 171, *175*
Liebelt, L. S., 55, *60*
Lieberman, D., 137, 139, 140, *146*, 223, *226*
Lieberman, H., 445, 448, *457*, *458*
Lieberman, J. S., 832, *852*
Liebrand, W. B. G., 138, *145*
Light, A., 270, *293*
Light, P., 769, 772, 773, 768, 769, *774*, 795, *808*
Lilienthal, M. G., 628, *637*
Lim, J. T. A., 29, *39*
Limbourg, Q., 1119, *1124*, *1125*
Lin, J., 1036, 1037, *1038*, *1039*
Lin, M. L., 310, *312*, *391*, 835, 843, 844, 847, 848, *851*, *852*
Lin, P., 1251, 1255, *1261*, *1262*
Lin, X., 440, *458*, 527, *543*
Lincoln, J. E., 207, 209, *225*
Lincoln, M., 889, *892*
Lindberg, D. A. B., 666, *677*
Linde, C., 1245, *1264*
Lindeman, R. W., 890, *893*
Linden, G., 442, 448, *456*, *457*

Lindenbaum, L. E., 29, *39*
Lindner, H., 219, *226*
Ling, D., 384, *429*
Ling, J., 219, *227*
Linton, M. A., 1033, 1034, *1039*
Linzer, M., 674, *677*
Lipinski, T. A., 1252, *1264*
Lips, H. M., 764, 765, *775*
Lipscomb, J., 163, 167, *173*, *175*, 426, *429*
Lipson, H. F., 547, *556*
Liscombe, J., 440, 451, *457*
Lisetti, C. L., 85, *90*, 439, 451, *457*
Liske, D., 29, *40*
Lisogurski, D., 838, *851*
Litman, D. J., 439, *457*, *458*
Little, J., 182, *196*, 834, *852*
Little, R., 1119, *1124*
Little, S., 218, *226*
Littleton, K., 768, 769, 772, *774*, 795, *808*
Littleton, M., 218, *225*
Littlewort, G., 451, *455*
Littman, J., 912, *915*
Littman, M., 581, *600*
Litwin, M., 1161, *1169*
Liu, J., 863, *868*
Liu, Y., 58, *62*, 574, *602*
Livingstone, C., 858, *867*
Lizza, C. S., 583, 591, *598*, *599*
Ljungberg, M., 735, *739*
Ljungstrand, P., 224, *225*, 285, *292*, 474, *486*
Llewellyn-Thomas, H. A., 669, *677*
Lloyd, D., 30, *41*
Lloyd, M., 764, *775*
Lloyd, P. A., 801, *809*
Lobach, D., 879, *882*
Localio, A. R., 667, *677*
Lock, S., 270, 277, *293*
Lockelt, M., 414, *431*
Lockhart, R. S., 45, 53, *59*, *60*
Lockwood, L. A. D., 993, *1015*, 1048, *1058*
Loewenstein, D. A., 782, *790*
Lofqvist, A., 890, *893*
Loftus, E. F., 54, *60*, 578, *599*
Logan, B., 83, *91*
Logan, G., 576, *599*
Lohman, D. F., 813, *828*
Loike, J. D., 781, *792*
Long, J. A. C., 14, *23*, 204, *226*
Long, S., 192, *196*
Longman, D. J. A., 70, *74*
Longstaff, T. A., 547, *556*
Lonnqvist, P., 262, 263, *264*, 1254, *1262*
Looi, C., 445*n*, *457*
Loomis, A., 843, 849, *852*
Lopes, C., 170, *172*, *196*
Lopez, A. D., 703, *718*
Lopez-Jaquero, V., 1124, *1125*
LoPresti, E. F., 835, 838, 843, 845, 847, *851*, *852*
Lorensen, B., 1033, *1039*
Lorenzo, M., 232, *245*
Loring, B., 1143, *1148*
Lorr, M., 87, *91*
Lortie, C., 33, *41*
Losch, A., 675, *677*
Louden, R. B., 1245, *1264*
Lourenço, O., 1247, *1264*
Lovejoy, T., 944, *947*, 967, *986*
Lovstrand, L., 1066, *1079*
Low, R., 420, 422, *430*
Lowe, C., 608, 610, *618*
Lowe, D., 182, *197*, 407, *411*
Lowe, K., 420, 422, *432*
Lowe, R. K., 576, *599*
Loxley, S., 251, *263*
Loy, M., 1033, *1038*
Loyall, A. B., 83, *89*
Lozier, J., 844, *850*
Lu, C. H., 36, *40*, 50, *60*, 606, *619*
Lucas, A., 735, *739*
Luck, R., 1065, *1079*
Luczak, H., 122, *131*, 202, 217, 218, 219, 223, 224, *226*, *227*
Lüder, E., 213, *226*
Ludolph, F., 1021, *1039*
Ludvigsel, M., 799, *807*
Luettin, J., 417, *429*
Luff, P., 549, *556*
Lukowicz, P., 182, *196*, 890, *892*
Lumsden, J., 262, *263*
Lund, A. M., 9, *23*
Lund, E., 1155, *1169*
Lundberg, G. D., 670, *678*
Lundman, D., 865, *867*
Lundmark, V., 548, *556*
Lunsford, R., 414, 420, 421, 422, *431*, *432*
Lunt, P., 647, *655*
Luo, Y., 582, *599*
Lurey, J. S., 583, *599*
Lusted, H. S., 841, *851*
Lustina, M. J., 770, *773*
Lutters, W., 651, *656*
Luyten, K., 1123, *1124*
Lykke-Olesen, A, 799, *807*
Lynch, C. I., 446, *458*, 770, *775*
Lynch, P., 652, *655*
Lynn, K. M., 767, 772, *774*
Lyon, C., 846, *850*
Lyon, D. R., 31, *39*
Lyons, J. L., 33, 35, 36, 37, 38, *40*, *41*
Lyons, K., 168, *173*, 299, 300, 303, 310, *311*, *312*
Lyons, M., 85, *90*, 843, *850*
Lyotard, J.-F., 1064, *1079*

M

Ma'Luf, N. I., 667, 668, *676*
Ma, J., 890, *893*
Maass, S., 1046, *1059*
Macadam, D. L., 207, *226*
MacDonald, J., 415, 424, *430*
MacEachren, A. M., 521, 522, 523, *543*
MacFarlane, S., 797, 800, *808*
MacGregor, D., 577, *599*
MacGregor, J., 55, *60*
Macgyvers, V., 844, *851*
Machrone, B., 139, *146*
MacInnis, D. J., 138, *146*
MacIntyre, A., 1245, *1264*

MacIntyre, B., 310, *311*, 1037, *1039*
Mack, R. L., 18, *23*, 498, *505*, 575, *599*, 1049, 1052, 1053, *1057*, *1058*, *1059*, 1181, *1189*
Mackay, W. E., 547, *556*, 797, *807*, 1018, 1022, 1023, 1024, 1027, 1028, 1030, 1031, 1036, *1038*, *1039*, 1250, 1259, *1264*
Macken, E., 1244, *1265*
MacKenzie, C. L., 29, 37, *40*
MacKenzie, I. S., 46, *60*, 162, 163, 165, 166, 168, 169, *172*, *173*, *175*, 184, 194, *197*, 844, *851*
Mackey, K., 472, *486*
Mackie, D., 764, *775*
Mackie, J. L., 1246, *1264*
Mackinlay, J. D., 162, *173*, 515, 517, 517*t*, 518, 523, 538, *542*, *543*, 566, 567, *571*, *572*
Mackworth, N. H., 45, *60*, 582, *599*
MacLean, A., 21, *23*, 1049, 1057, *1058*, 1066, *1079*
MacLean, K. E., 168, *176*, 420, 422, *432*
Macmillan, N. A., 45, *60*
Macredie, R., 57, *58*
Madani, O., 581, *600*
Madden, M., 604, *619*
Mader, S., 606, *619*
Madhavapeddy, A., 270, *294*
Madigan, S. A., 492, *506*
Madison, D. S., 909, *915*
Madsen, K. H., 1025, *1038*, 1049, *1058*, 1065, 1067, 1068, 1072, 1075, *1078*, *1079*, 1250, *1264*
Maeda, F., 549, *557*
Maeda, T., 230, *244*
Maes, P., 170, *175*, 180, *197*, 435, 436, 446, 447, 448, 454, *457*, *458*
Magee, J. S., 549, *556*
Magee, L., 51, *59*
Magill, R. A., 70, *75*
Maglio, P., 168, 170, *173*, *174*, 568, *572*
Maguire, T. O., 139, 140, *146*
Magyar, R., 167, *175*
Mahajan, M., 862, *868*
Mahalingam, S., 181, *198*
Mahapatra, M., 1246, *1265*
Maher, M. L., 1065, *1079*
Maibach, E., 81, *91*
Maiden, N., 1054, 1055, *1057*
Maier, C., 1064, *1078*
Mailloux, C. A., 1251, *1261*
Mainwaring, S. D., 277, *293*, 552, *554*, 642, *655*
Maio, D., 182, *198*
Mair, G., 625, *635*
Majercik, S., 581, *600*
Majoe, D., 1098, 1102*f*, *1105*
Majumder, A., 623, *637*
Makhoul, J., 415, *428*
Malamuth, N., 79, *90*
Maldari, M. S., 139, *146*
Maldonado, H., 87, 88, *90*
Malin, B., 643, 654, *657*
Malkewitz, R., 843, *851*
Malkin, J., 181, *196*
Malkin, R., 428, *429*
Malone, T. W., 136, *146*, 501, *505*, 548, *556*, 683, 684*t*, *700*, 746, *759*, 767, 772, *774*, 1255, *1264*
Maloney-Krichmar, D., 610, *619*, 781, *791*
Maltoni, D., 182, *198*
Malvar, H. S., 181, *198*
Mambrey, P., 1077, *1081*
Manaris, B., 837, 844, *851*
Manasse, B., 845, *851*
Mancini, C., 864, *869*
Mancini, L., 647, *655*
Mandal, A. C., 319, *325*
Mandler, G., 649, *657*
Mandler, J. M., 818, *828*
Mandryk, R. L., 37, *40*, 171, *176*, 683, *700*
Manes, D., 714, *718*
Mangione, C. M., 858, *869*
Manji, K. A., 250, 263
Manke, S., 426, *429*
Mankoff, J., 644, *657*, 1033, *1038*, 1070*n*, *1078*, 1254, *1264*
Manligas, C., 252, *264*
Mann, A. M., 666, *676*
Mann, W. C., 399, *411*
Manning, G. K., 1212, *1219*
Manning, H., 956, *963*, 980, *987*, 1173, *1189*
Manny, C. H., 642, *657*
Mansfield, A. F., 612, *618*
Mansur, D. L., 250, 252, 261, *264*
Mantei, M. M., 643, *657*
Manthouli, M., 137, *145*
Marable, L., 143, *145*, 652, *656*
Marcella, S., 1077, *1078*
March, A., 15, *23*
Marchionini, G., 527, *543*, 1254, *1263*
Marcoen, A., 54, *61*
Marcus, A., 17, *24*, 356, 360, 361, 363, 375, 376, 377, *378*, *379*, 1081, 1087, *1105*
Marcus, G., 968, *986*, *987*
Marentakis, G., 255, 262, *264*
Mariani, J. A., 274, 277, *292*, *293*, 841, *849*
Marila, J., 259, *264*
Mark, G., 492, *506*, 1254, *1261*
Markley, D., 1172, *1189*
Marklin, R., 168, *175*
Markoff, J., 2, *24*
Markopoulos, P., 800, 801, *807*, *808*
Marks, I. M., 139, *145*
Marks, S. J., 474, *485*, 1177, 1184, *1188*
Markus, M. L., 546, 551, *556*
Marmasse, N., 169, *176*
Marquis-Faulkes, F., 826, *828*
Marquise, 498, *506*
Marsden, G., 171, *174*
Marsh, S., 583, *600*, 652, 653, *657*
Marshall, C., 139, 140, *146*
Marshall, J. R., 142, *145*, 279, *292*, 652, *656*, 705, *719*
Marshall, R. S., 764, *774*
Marsic, I., 427, *432*
Marteniuk, R. G., 28, 29, *40*
Marti, S., 169, *176*, 475, *487*, 548, *557*
Martin, B., 445, *457*
Martin, C., 394, 400, *410*
Martin, D. L., 427, *430*
Martin, F., 485, *487*, 804, *808*
Martin, J., 6, *24*
Martin, J.-C., 414, *428*
Martin, K., 1033, *1039*
Martin, R., 296, 298, *311*, *312*
Martin, T., 308, *312*
Martinez, A. M., 85, 90
Martinussen, M., 632, *637*

Maruyama, S., 168, *175*
Marwood, D., 647, *656*
Marx, L., 1242, 1243, *1266*
Marx, M., 389, *391*, 862, *870*
Masarie, F., 595, *597*
Mase, K., 180, *197*, 426, *429*, 444, 445*f*, *457*
Masera, M., 1254, *1263*
Mashyna, M. M., 1178, *1188*
Maslow, A.H., 82, *90*, 901, *915*, 1145, *1148*
Mason, R., 608, *619*
Mason, S., 838, 841, 842, *849*, *850*
Mason, S. G., 850, *851*
Massaro, D. W., 415, 417, 420, 424, *429*, *430*
Massie, T., 231, *245*
Massimino, M., 624, *637*
Mastaglio, T., 570, *572*, *593*, *598*
Masten, D., 974, *987*
Matchen, P., 499, *506*
Matejka, J., 171, *174*
Matessa, M., 58, 107, 804, *111*, *807*, 1198, *1207*
Mathan, S., 1228, *1238*
Matherwson, N., 643, *656*
Mathiak, K., 683, *700*
Mathiassen, L., 1250, *1264*
Mathieu, J. E., 72, 73, *75*
Matias, E., 844, *851*
Matschulat, H., 212, *226*
Matsushita, N., 171, *175*, 181, *197*
Matsushita, Y., 549, *557*
Mattelmäki, T., 977, *987*, 1070, *1079*, *1080*
Matthews, G., 116, 121, 127, 128, 129, *131*, *132*, 786, *792*
Matthews, P. M., 1234, *1238*
Matthews, T., 1253, *1262*
Matthiessen, L., 1062, 1063, *1078*, *1079*
Mattis, J., 517, 536, *542*, *543*
Maury, S., 259, *264*
Mauve, M., 288, *293*
Mavor, A. S., 109, *112*, 623, 626, *636*, 735, *740*
Maxfield, A., 665, *676*
Maxion, R., 648, *657*
May, J. G., 285, *292*, 403, *411*, 623, *637*
May, K., 168, *173*
May, R. A., 541, *543*
Mayer, K., 709, *717*
Mayer, N., 1119, *1124*
Mayer, R. E., 69, 70, 71, *75*
Mayhew, D., 1270, *1277*
Mayhew, D. J., 64, *75*, 918, 919*f*, *926*
Mayhew, D. R., 707, 711*t*, *718*
Mayhew, G. E., 744, *759*
Mayhorn, C. B., 593, *602*, 782, 785, 786, *791*
Maylor, E. A., 33, *40*
Maynes-Aminzade, D., 442, *455*, 473, *486*
Mayser, C., 706, *718*
Mazalek, A., 474, 481, *486*
Mazur, S. A., 14, *23*
McAuliffe, J., 31, *40*
McBride, D., 1222, *1239*
McBride, R., 13, *23*
McCabe, T. J., 502, *506*
McCall, K., 550, *555*
McCalla, G., 445*n*, 457
McCann, R. S., 732, *739*
McCarthy, G., 47, *59*
McCarthy, J., 8, *24*, 583, *601*
McCarthy, J. D., 1254, *1265*
McCarthy, J. F., 1249, *1264*
McCarthy, K., 442, *457*
McCartt, A. T., 705, *718*
McCauley, C., 579, *600*
McCauley, M. E., 629, *637*
McCauley-Bell, P. R., 631, *637*
McClard, A., 1175, *1189*
McClelland, J. L., 47, 48, 49, *60*, *61*
McCloskey, M., 64, *75*
McConnell, R. L., 181, *198*
McCorduck, P., 13, *23*
McCormick, B. H., 517, *543*
McCormick, E. J., 207, 209, *227*, 248, *264*, 723, 724, *739*, 1228, *1239*
McCowan, I., 428, *429*, *430*
McCoy, E., 574, 576, 580, 583, 596, *601*
McCoy, K. F., 818, *828*, 837, 840, 845, *850*, *851*, 862, *869*
McCracken, D. D., 563, *572*
McCracken, W. M., 1054, *1057*
McCrae, R. R., 121, 127, *131*
McCreath, E., 435, 452, *456*, *457*
McCrickard, D. S., 583, *597*, 1056, *1058*, *1059*
McCurdy, M., 1198, *1207*
McDaniel, R. G., 498, *506*, 1035, *1039*
McDaniel, S. E., 276, *293*, 549, *556*
McDermott, J., 436, *457*, 847, *851*
McDermott, K. B., 54, *61*
McDonald, C. J., 663, *677*
McDonald, D. W., 445, 448, *457*
McDonald, H. E., 81, *90*
McDonald, J. E., 55, *60*
McDonald, M., 142, *146*
McDonald, S., 877, 878, *882*
McElligott, J., 796, *808*
McEvoy, L. K., 1234, *1239*
McEvoy, S. P., 705, *718*
McFadden, S., 220, *226*
McFarland, A. L., 734, *738*
McFarlane, D., 168, *174*
McGarr, N. S., 890, *893*
McGee, D. R., 414, 417, 419, 428, *429*, *430*
McGee, J. P., 735, *740*
McGee, L. A., 723, *739*
McGee, M., 1138, 1146, *1148*, 1172, *1189*
McGee, M. R., 839, *851*, 859, *869*
McGill, M. E., 517, *542*
McGinty, L., 442, *443*
McGlone, F., 30, *41*
McGookin, D. K., 258, *264*
McGovern, G., 993, *1016*
McGrath, J. E., 547, 548, *555*, *556*, *557*
McGrath, M., 415, *430*
McGreal, M. J., 669, *677*
McGreevy, M. W., 735, *739*
McGrenere, J., 436, 453*f*, 453, *455*, *456*, *457*, 879, *882*, 1077, *1078*, *1080*
McGuire, R., 297, *312*
McGuire, T. W., 548, *557*
McGurk, H., 415, 424, *430*
McIntosh, B. B., 724, *738*
McIntyre, R. M., 72, *75*
McIsaac, P., 841, *850*, *851*
McKenna, S. J., 826, *828*
McKerlie, D., 1057, *1058*

McKevitt, P., 437*n*, 456
McKinlay, A., 817, *828*
McKinley, J. L., 1244, *1265*
McKnown, J., 796, *807*
McLachlan, P., 420, 422, *432*
McLagan, P., 1062, *1080*
McLaughlin, A. C., 786, *791*
McLaughlin, M. P., 733, *737*
McLeod, A., 415, 420, *430*
McLeod, P. L., 546, 548, 550, *556*
McLoone, H., 168, 169, *175*
McLoughlin, C., 610, *619*
McLuhan, M., 604, *619*
McManus, B., 275, *293*
McManus, C., 219, *227*
McMillan, G. R., *1195*, *1207*
McMillan, W. W., 863, *869*
McMullan, B., 331, *353*
McMurray, J. E., 674, *677*
McNair, D. M., 87, *91*
McNamara, D. S., 71, 72, *75*
McNee, S. M., 448, *458*
McNeely, W., 231, *245*
McNeill, D., 181, *197*, 415, 420, 422, 423, 424, *430*
McNish, M. M., 69, *76*
McNulty, M., 1137, 1143, *1149*
McPhail et al., 1068
McPhail, B., 1067, 1068, *1080*
McQuire, M., 579, *602*
McRoy, S., 439, *456*
McRuer, D., 727, *739*
McTear, M. F., 796, *808*
Mead, A., 633, *637*
Mead, S. E., 782, 786, 787, *791*
Meader, D. K., 548, *557*
Meadows, C., 647, *657*
Means, B., 935, *947*
Medin, D. L., 1043, *1059*
Médini, L., 1026, 1039
Medl, A., 427, 432
Medlock, M. C., 751, 752, 753*t*, 754*f*, *759*, 1146, *1148*, 1172, *1189*
Medsker, G. J., 1160, *1169*
Meece, J. L., 771, *774*
Meegan, D. V., 29, 34, 35, *40*, *41*
Meehl, P., 502, *506*
Meggs, P. B., 348, *353*
Meghan R. 1141, *1148*
Mehandjiev, N., 406, *410*
Mehl, W. M., 1062, *1078*, 1250, *1262*
Mehr, E. B., 855, *869*
Meier, M., 475, *487*
Meier, N., 211, *227*
Meier, U., 417, *430*, *432*
Meiskey, L., 1175, *1189*
Meister, D., 3, *24*, 574, 582, *600*, 1133, 1136, *1148*
Mekhjian, H. S., 667, *676*, *677*
Mello, S., 993, *1016*
Mellor, B., 846, *850*
Melton, R. J., 81, *90*
Melzer, J., 310, *312*
Memon, N., 649, *658*
Mendel, M. B., 724, *739*
Mendoza, J. E., 33, *41*
Menozzi, M., 217, 222, *226*
Mens, T., 1117, *1125*
Menzies, D., 626, *637*
Mercatante, T. A., 631, *638*
Mercer, J., 735, *739*
Mereu, S. W., 863, *869*
Merikansky, A., 858, *869*
Merkel, C. B., 1065, *1080*
Merritt, E. A., 626, *637*
Merton, R. K., 766, *774*
Mervis, C. B., 1043, *1059*
Merwin, D., 730, *740*
Meskill, J. A., 1175, *1189*
Messer, D., 768, 769, *774*, 795, *808*
Messerschmitt, D. G., 547, *555*
Messeter, 1072, *1073*
Messeter, J., 1077, *1079*
Messina, P., 547, *555*
Metaxas, D., 890, *893*
Metcalfe, B., 604, *619*
Metcalfe, R., 134, *145*
Metlay, J. P., 667, *677*
Metzger, M. J., 45, 48, *61*, 142, 143, *146*
Metzger, U., 582, 595, *600*
Mewhort, D. J. K., 56, *59*
Meyer, D. E., 29, *40*, 47, 50, 56, 58, *60*, 104, 104*f*, 105, 106, 107, 110, *112*, 503, 575, *599*, 1199, 1202, 1203, *1207*
Meyer, J., 1031, 1033, *1038*
Meyer, S., 643, *657*
Meyers, B., 170, *174*, 188, *196*, *197*
Meyers, D., 575, *600*
Micarelli, A., 442, *457*
Michaels, C. F., 57, *60*, 847, *851*
Michalewicz, Z., 580, *600*
Michalski, R., 567, *572*
Michie, D., 580, *600*
Michon, J. A., 702, 706, *718*
Michotte, B., 1124, *1125*
Midkiff, A. H., 736, *739*
Midtland, K., 711*t*, *718*
Mikhak, B., 473, *486*
Mikkelson, N., 941, *947*
Mikulka, P., 582, *598*, *600*
Mikulka, P. J., 120, *132*, 1227, *1238*, *1239*
Milberg, S. J., 643, *657*
Miles, M. B., 940, *947*
Milewski, A. E., 1253, *1264*
Milgram, P., 163, 167, *176*
Milham, L. M., 633, *637*
Millen, D. R., 29, *39*, 549, *556*, 908, 908*t*, 909, *915*
Millen, J., 647, *657*
Miller, C., *574*, 581, *600*, 802, *808*, 1233, *1238*
Miller, D., *987*, *1148*
Miller, D. C., 1195, *1207*
Miller, G. A., 51, 52, *60*, 494, *506*, 575, *600*
Miller, J., 47, 48, *60*, 966, 975, *987*
Miller, J. B., 1246, *1265*
Miller, J. D., 840, *851*
Miller, J. K., 1233, 1246, 1248, 1255, 1257, 1259, *1262*
Miller, J. R., 1178, *1188*
Miller, K., 1167, *1169*, 1248, *1263*
Miller, L. T., 794, *808*
Miller, P., 593, 595, *600*
Miller, R., 595, *597*, 651, *657*, 832, *851*
Miller, R. A., 582, *600*

Miller, R. C., 1035, *1039*
Miller, S. T., 415, *428*, 794, 802, *866*
Miller-Albrecht, S. A., 787, *791*
Miller-Jacobs, H. H., 1135, *1148*
Millett, L. I., 279, *293*, 1251, 1255, 1256, *1262*, *1264*
Millians, J., 784, *792*
Milligan, T., 643, *657*
Mills, C. B., 71, 72, *75*
Mills, M., 796, *807*
Milton, J. L., 724, *738*
Minassian, S., 435, *456*
Mine, M., 167, *175*
Minneman, S., 551, *556*
Minocha, S., 1046, 1055, *1059*
Minsky, M., 233, *245*
Minuto, A., 217, *225*
Miranda, E. R., 252, 253, *264*
Mirchandani, N., 878, *882*
Mirenda, P., 816, *827*
Mishra, D., 879, *882*
Miskelly, C., 1065, 1069, 1070, *1077*
Mital, A., 574, *600*
Mitchell, B. M., 547, *555*
Mitchell, C. M., 574, 575, 582, *599*, *600*, *601*
Mitchell, K., 185, 186, *196*
Mitchell, R. E., 886, *893*
Mitchell, T. M., 192, *197*, 436, *457*
Mithal, A. K., 162, 163, 166, *173*
Mitra, A., 764, *774*
Mitrani, V. B., 782, *790*
Mitropoulos-Rundus, D., 1136, 1137, *1148*
Mitrovic, A., 445, 446, *457*
Mitsch, D., 1054, *1058*
Mitsopoulos, E., 278, *293*
Mittleman, D. D., 15, *24*
Mittman, R., 674, *677*, 873, *881*
Mival, O., 826, *827*
Miyake, N., 553, *556*
Miyao, M., 219, 224, *226*
Miyasato, 232, *245*
Mobasher, B., 450, *457*
Mochon, C., 170, *174*
Mock, J. E., 1212, 1215, *1219*
Mockus, A., 73, *75*, 552, *556*
Moeykens, B. A., 873, *882*
Moffatt, K., 879, *882*, 1077, *1080*
Moffitt, K., 310, *312*
Moffitt, M. D., 436, *456*
Mogensen, P., 1065, *1080*
Moggridge, B., 137, *146*
Moghaddam, B., 185, *198*
Mohan, D., 703, *718*
Mohd-Nasir, N., 171, *174*
Molenaar, S., 669, *677*
Molich, R., 1139, 1140, 1141, *1148*, 1175, *1189*, 1255, 1259, *1264*
Moline, J., 634, *637*
Mollaghasemi, M., 630, *638*
Moller, J., 576, *599*
Molnar, G. E., 833, *851*
Moloney, K. P., 854, 859, 861, *867*, *868*, *869*
Monaco, V., 873, *881*
Monk, A. F., 17, *22*, 194, *196*, 549, *555*, *556*, 1067, 1068, *1080*
Monroe, J., 168, *175*
Monrose, F., 649, *656*, *657*
Monsell, S., 50, 56, *60*, *61*
Montaniz, F., 1181, *1189*, 1276, *1277*
Monteiro, K. P., 81, *89*
Montemayor, J., 797, *806*, 1065, *1078*
Monty, R. A., 723, *739*
Moody, P., 435*n*, *456*, 548, *558*
Moon, B. D., 541, *543*
Moon, R., 874, *882*
Moon, Y., 82, *90*, 139, 140, *146*, 178, *197*
Moor, J. H., 1247, 1248, *1264*
Mooraj, Z., 417, *429*, 840, *850*
Moore R. J., 608, *618*
Moore, A., 568, 569, *571*, *572*
Moore, B. A., 81, *89*
Moore, B. C., 251, 252, *264*
Moore, J. S., 833, 842, 849, *851*
Moore, K., 744, *759*
Moore, M. M., 842, 849, *850*, *851*
Moore, R. J., 972, *987*
Moore, S. M., 782, *791*
Moorman, C., 138, *146*
Morahan-Martin, J., 765, *774*
Moran, D. B., 395, 427, *429*, *430*
Moran, R. T., 365, *377*
Moran, T., 165, 166, 171, *173*, *175*, 274, 277, 286, *292*, *293*, 1057, *1059*, 1066, *1079*, 1130, *1149*, 1195, 1198, 1204, 1205, *1207*
Moran, T. P., 7, 12, 16, *22*, *23*, 44, *59*, 65, 66, 68, *75*, 94, 96, 97, 97*f*, 108, *111*, 277, *292*, 395, *410*, 490, *505*, 534, *542*, 551, *556*, 575, *597*, 714, *716*, 839, *850*, 1049, *1058*, *1207*, 1204, 1214, *1218*
Moray, N. P., 66, *75*, 120, 124, *131*, 148, *157*, 583, *600*
Morch, A. I., 593, *598*
Mørch, A. I., 1067, *1080*
Morency, L.-P., 417, *430*
Moreno, R., 69, 70, *75*
Morgan, D., 389, *390*
Morgan, M. E., 826, *827*, *828*
Morgan, R. L., 70, *76*
Mori, G., 1117, *1125*
Morimoto, C., 181, *199*, 414, 417, *430*, *432*
Morkes, J., 83, *91*, 139, *146*
Morley, S., 863, *869*
Morrell, R. W., 781, 782, 785, 786, *791*, *792*
Morris, C. D., 53, *60*
Morris, C. S., 633, *637*
Morris, J., 553, *556*
Morris, M. W., 548, *555*
Morris, P., 1254, 1255, *1263*
Morris, T., 1249, *1263*
Morrison, J. B., 108, *111*
Morrison, J. G., 1223, *1238*
Morse, D. R., 191, *196*
Mortensen, M., 583, *598*
Morton, J., 495, *506*
Moscovitch, M., 51, *61*
Moses, P., 841, *850*, *851*
Mosher, A., 966, *986*, 1066, *1077*
Mosier, J. N., 14, *24*, 551, *556*
Mosier, K., 576, *601*
Moskowitz, H., 629, *638*
Motavalli, M., 862, *867*
Mottet, D., 166, *174*
Mottram, M., 786, *790*
Motz, F., 218, *227*
Mount, M. K., 127, *130*

Mountford, S. J., 735, *739*
Mourant, R., 627, *638*
Mousavi, S. Y., 420, 422, *430*
Movellan, J. R., 451, *455*
Moy, G., 233, *245*
Mozer, M. C., 191, *197*
Mrazek, D., 934, *946*
Mucherah, W.-M., 765, *774*
Mudge, D., 647, *658*
Muehlenhard, C. L., 138, 140, *146*
Mueller, S., 110, *112*
Mueller-Spahn, F., 138, 145, *145*
Muir, B., 583, *600*
Mukopadhyay, T., 548, *556*
Mulgund, S., 624, *637*
Mullen, Jr., M., 168, *174*
Muller, M. J., 549, 552, *556*, 1025, *1039*, 1045, 1046, 1049, *1059*, 1062, 1063, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1075, 1077, *1079*, *1080*, *1081*, 1175, *1189*, 1250, 1253, *1261*, *1264*
Muller, M. K., 1052, *1059*
Muller, R. L., 139, *146*
Müller, C., 440, *457*
Müller, H. J., 31, *40*
Müller-Tomfelde, C., 1112, *1125*
Mullert, N., 1067, *1079*
Mullet, K., 394, 403, 404, 409, *411*, 1065, *1081*
Multon, F., 414, *432*
Mumford, E., 6, *24*, 1062, 1063, 1065, *1080*
Mumford, M., 581, *600*
Muncher, E., 99, *112*
Munhall, K. G., 415, *432*
Munk-Madsen, A., 1065, 1074, *1079*
Muñoz, R., 1135, *1148*
Munteanu, E., 862, *867*
Munzner, T., 539, *543*
Murakami, T., 231, *245*
Murata, A., 784, *791*
Murdock, B. B., Jr., 45, *60*
Muresan, M., 713, *719*
Murphy, J., 873, *883*, 970, 974, *987*, 1071, *1079*
Murphy, L. L., 117, 122, *131*, 1145, *1148*
Murphy, M. D., 787, *791*
Murphy, M. R., 723, *739*
Murphy, R., 359, *379*
Murphy, S. B., 858, *867*
Murphy, S. T., 84, *91*
Murray, C. J. L., 703, *718*
Murray, I. R., 81, *91*, 219, *227*
Murray, J. H., 628, *637*
Murray, N., 86, *91*
Murray, P. J., 670, *677*
Murray, W., 218, *226*
Musacchio, R. A., 670, *678*
Musen, M. A., 668, *677*
Muss, T., 166, *174*
Muszak, J., 1136, 1137, *1148*
Muter, P., 224, *227*
Muth, E., 1226, *1238*
Mwanza, D., 617, *619*
Myers, B., 165, 169, 171, *173*, *175*, 840, 845, *851*, *852*, 1112, 1123, *1125*, 1153, *1169*
Myers, B. A., 2, *24*, 318, *325*, 498, *506*, 1032, 1035, *1039*
Myers, I. B., 674, *677*
Myers, J., 1074, *1078*
Mylonopoulos, N. A., 6, *24*
Mylopoulos, J., 1055, *1059*
Mynatt, C., 579, *600*
Mynatt, E. D., , 171, *172*, 256, 260, 261, 262, *264*, 863, *867*, *869*, 1245, 1249, 1256, 1255, *1264*, *1266*

N

Nabamoo, D., 318, *325*
Nachemson, A., 319, 320, *325*
Nachman, L., 179, *197*
Nader, L., 970*n*, *987*
Nadler, E., 117, *131*
Nagel, D. C., 582, *602*
Nagel, T., 1246, *1264*
Naghshineh, M., 1123, *1124*
Nah, F. F.-H., 6, *24*
Nahmens, I., 629, *638*
Nahrstedt, K., 273, *292*
Nair, S. N., 780, 781, 782, 785, 788, *790*, *791*, *792*
Nairne, J. S., 51, *60*
Nakajima, N., 231, *245*
Nakatsu, R. T., 583, *600*
Nakisa, M., 889, *892*
Namioka, A., 575, *601*, 797, 798, *808*, 982, *987*, 1062, 1063, *1081*
Nanetti, R., 1254, *1265*
Näpflin, U., 222, *226*
Napoli, J., 623, *636*
Narayanan, N. H., 403, 408, *411*
Nardi, B. A., 549, 550, 552, *556*, *557*, 966, 980, 982, *987*, 1048, *1059*
Nareyek, A., 581, *600*
Narin, F., 567, *572*
Narita, A., 475, *486*
Nason, S., 575, *600*
Nasoz, F., 439, 451, *457*
Nass, C., 79, 81, 82, 83, 84, 87, *91*, 137, 139, 140, *145*, *146*, 403, 404, *411*, 682, *699*, 859, *869*, 901, *915*, 1253, 1256, *1264*, *1265*
Nass, C. I., 140, *146*
Nau, D., 581, *598*
Naughton, K., 423, *430*
Naur, P., 1185, *1189*
Navon, D., 55, *60*, 123, 124, *131*, 579, *600*
Neal, J. G., 414, 415, 426, *430*
Neal, R., 186, *197*
Neale, D. C., 583, *597*, 1044, *1057*
Neale, H., 795, *808*, 1065, *1078*
Neale, M. A., 583, *598*
Nedell, W., 736, *738*
Neese, R. M., 79, *91*
Neggers, S. F. W., 35, 38, *39*
Negroponte, N., 8, *24*, 414, *430*
Neisser, U., 148, *157*
Nejdl, W., 441*n*, *455*
Nel, C., 1062, *1080*
Nelson, G., 101, *112*
Nelson, L. J., 771, *774*
Nelson, T., 5, *24*
Nelson, T. H., 570, *572*
Nelson, T. J., 207, 207*f*, 209, 214, 215, *226*
Neti, C., 414, 415, 417, 424, *430*, *431*
Nettle, D., 578, *598*
Neuhoff, J. G., 255, *264*
Neumann, N., 841, *849*

Neumann, P. G., 1252, 1259, *1264*
Neumann, R., 84, *91*
Neunschwander, E., 134, *145*
Neustaedter, C., 907, *915*
Newell, A., 7, 8, 12, 14, 16, 18, *22*, *23*, *24*, 44, 56, *59*, *60*, *75*, 94, 96, 97, 97*f*, 98, 100, 108, *111*, *112*, 151, *157*, 166, *173*, 286, *292*, 395, *410*, 490, 491, 534, *542*, 575, 579, 580, *597*, *599*, *600*, 714, *716*, 817, 818, 822, 823, 824, 825, 826, 827, *827*, 838, 839, 840, *850*, 872, 879, *882*, 1195, 1196, 1198, 1204, 1205, *1207*, 1214, *1218*
Newell, A. F., *827*, *828*, *849*, *850*
Newell, B. R., 503, *505*, *506*, *507*
Newhagen, J., 81, *91*
Newman, D., 802, 805, *808*
Newman, M. W., 1036, *1039*
Newman, S. E., 977, *987*
Newport, E., 886, *893*
Newstetter, W. C., 1253, *1266*
Ng, J. D., 644, *656*
Nguyen, L. K., 633, *637*
Nguyen, T. C., 667, *676*
Nicholls, J. G., 771, *774*
Nichols, J., 840, *851*
Nichols, M. E. R., 30, *41*
Nicholson, J., 769, 770, 773, *774*
Nickell, E., 608, *618*
Nickerson, R., 582, *600*
Nicolai, P., 1212, *1219*
Nicoletti, R., 36, *40*, *41*, 50, *61*
Nicolle, C. A., 560, *571*
Nie, J., 567, *572*
Niedenthal, P. M., 81, *91*
Nielsen, J., 18, *24*, 142, *146*, 153, *157*, 194, *197*, 361, 377, *377*, *378*, 398, 407, *411*, 501, *506*, 751, 754, 755, *759*, 785, *791*, 795, 797, 800, *807*, *808*, 824, *828*, 876, *882*, *987*, 992, *1016*, 1052, *1059*, 1109, *1125*, 1133, 1138, 1139, 1142, *1148*, 1152, 1155, *1169*, 1175, 1178, 1179, 1180, 1181, 1184, 1185, *1189*, 1244, *1264*, 1269, 1276, *1277*
Nielsen, P. E., 96, *112*
Nielsen, T., 171, *175*
Nielson, J., 560, 563, *572*
Niemi, R. G., 1245, *1261*
Nies, K., 646, *656*
Nijholt, A., 549, *558*
Nilsen, E., 51, *61*, 714, *718*
Nilsson, B., 218, *225*
Nilsson, J., 644, *657*
Nilsson, S. E. G., 859, *869*
Nilsson, U. L., 859, *869*
Nimmo-Smith, I., 219, *227*
Nisbett, R. E., 56, *59*, 66, *75*, 138, *146*, 365, *378*, *379*, 578, *599*, 746, *759*
Nishioka, A., 804, *807*
Nishiyama, K., 316, *325*
Nisonen, E., 1067, *1080*
Nissen, M. J., 31, *40*
Nissenbaum, H., 1247, 1248, 1251, 1253, 1255, *1262*, *1264*
Nix, D., 796, *808*
Nixon, B., 1055, *1059*
Njie, V., 873, *881*
Noakes, P., 474, *485*
Noble, D. D., 1243, *1264*
Noble, A., 1069, 1070, 1074, *1080*
Nock, H., 414, *430*
Nodari, E., 708, *717*
Nohria, N., 548, *557*
Noma, H., 232, *245*
Noonan, T. K., 45, *60*
Noorlander, J., 669, *677*
Norberg, A. L., 7, 13, *24*
Nordo, E., 1200, *1206*
Norman, D. A., 14, 16, 19, *23*, *24*, 65, *75*, *157*, 124, *131*, 148, 149, 167, *175*, 194, *197*, 396, 403, *411*, *457*, 446, 485, *486*, 496, *506*, 517*t*, 576, 578, 582, *600*, *602*, 680, 681, 682, 683, 685, 687, 689, 684*t*, *700*, 726, *739*, *740*, 901, 901*n*, 907, *915*, 929, *947*, 1020, *1039*, 1047, 1048, 1051, *1059*, *1189*, 1197, *1207*, 1244, *1264*
Norman, K. L., 83, 87, *89*, *91*, 1138, *1147*
Normark, M., 733, *738*
Noro, K., 323, *325*, 1062, 1063, *1080*
North, M. M., 625, *637*, *638*
North, S. M., 625, 637
Norvig, P., 437*n*, *456*, 580, *601*
Not, E., 452, *457*
Noth, M., 1255, *1266*
Nowak, G. J., 138, *146*
Nowakowski, C., 705, 709*t*, *717*, *718*
Nowicki, G. P., 81, *90*
Nowicki, J., 1276, *1277*
Noy, Y. I., 702, *718*
Nunamaker, J. F., 15, 18, *23*, *24*, 550, *557*
Nusbaum, H., 385, *390*
Nussbaum, M. A., 360, *378*
Nutt, G., 546, *555*
Nutter, E. H., 1069, 1074, 1075, *1080*
Nygaard, K., 1062, *1080*

O

O'Conaill, B., 548, *557*
O'Connell, T., 193, *196*
O'Connor, A. M., 668, *677*
O'Connor, K. M., 547, *555*, *556*
O'Connor, P., 669, *678*
O'Day, V. L., 1245, *1264*
O'Donnell, R. D., 120, *131*, 501, *506*
O'Hara, K. P., 71, 72, *75*
O'Hara-Devereaux, M., 547, *557*
O'Hare, E. A., 796, *808*
O'Malley, C., 795, *808*
O'Modhrain, S., 474, *486*
O'Neill, D. K., 1249, *1265*
O'Neill, J. E., 7, 13, *24*, 501, *506*
O'Shea, T., 138, 139, *145*, 256, *264*, 276, *292*
Oakley, B. W., 13, *24*
Oakley, I., 168, *175*, 839, *851*, 859, *869*
Oates, R., 666, *677*
Oatley, K., 79, *91*
Oba, H., 474, *487*
Oberman, F., 1200, *1206*
Obradovich, J. H., 598, *601*
Ockerman, J., 300, *312*
Odobez, J.-M., 428, *429*
Oehme, O., 202, 217, 218, 219, 224, *226*, *227*
Oetjen, S., 219, 220, 221*f*, 222, *226*
Ogden, B., 862, *868*
Ogden, W. C., 31, *40*
Ogozalek, V. Z., 788, *791*, 812, *828*
Ohlsson, S., 446, *457*
Oinas-Kukkonen, H., 560, *571*
Oja, S., 713, *719*

Okada, K., 549, *557*
Okebukola, P. A., 765, *774*
Okuno, H. G., 439, 440*f*, *456*
Olami, Z., 49, *61*
Oldham, G. R., 1152, *1168*
Olefsky, K., 767, *774*
Olfman, L., 547, *557*
Olgyay, N., 361, *378*
Oliver, D., 581, *600*
Oliver, N., 165, 170, *176*, 182, *197*, 427, *430*
Oliviè, H., 864, *869*
Olmos, O., 735, *739*
Olsen, D. R., 171, 172, *173*, *175*, 181, 186, *196*
Olsen, E., 419, 424, *431*
Olsen, R. A., 181, 186, 196
Olson, G. M., 167, *175*, 546, 548, 549, 550, 551, 552, 553, 554, *555*, *556*, *557*, *558*, 574, 576, 583, *600*, 1254, *1261*, *1264*, *1266*
Olson, J., 1249, 1254, *1261*
Olson, J. R., 65, *74*, 110, *175*, 714, *718*
Olson, J. S., 546, 548, 549, 550, 551, 552, 553, 554, *555*, *556*, *557*, *558*, 574, 576, 583, *600*, 1249, 1254, *1264*, *1266*
Olson, M. H., 277, *293*, 554, *557*
Olufokunbi, K. C., 617, *619*, 1250, *1264*
Omata, S., 854, 865, *867*
Omori, M., 224, *226*
Ona, Y., 365, 378
Onayade, A. A., 1250, *1264*
Oniszczak, A., 163, *175*
Ono, Y. A., 214, *226*
Oosterholt, R., 795, *808*
Opdyke, D., 634, *638*
Öqvist, G., 224, *225*
Oransanu, J., 574, 583, *600*, *601*
Oransanu, R., 583, *599*
Orbanes, J., 194, *197*
Ordy, J. M., 625, 630, *637*
Oritz, A., 858, *868*
Orlikowski, W. J., 552, *557*, 1054, *1059*, 1244, 1248, *1265*
Orman, L., 673, *677*
Ormerod, T. C., 153, *157*
Ormrod, J., 576, *600*
Ormsby, G., 1102, *1105*
Ornstein, S. M., 666, *677*, *678*
Oron-Gilad, T., 118, 120, 123, 127, 128, 129, 132, *131*
Orr, A. L., 854, 858, *869*
Orr, J., 1055, *1059*, 1066, *1080*
Ortengreen, R., 319, 320, *324*
Orth, M., 474, *486*
Ortony, A., 79, 83, *91*, 405, *411*, 901, *915*
Osborn, A., 1022, *1039*
Osborn, R., 662, *678*
Oseitutu, K., 384, *391*, 835, 838, 844, 847, *851*, *852*
Osgood, C. E., 632, 633, *637*
Oshagan, H., 84, *91*
Osipovich, A., 142, *145*, 279, *292*, 652, *656*
Osman, A., 49, *60*
Osman, I., 580, *600*
Ostberg, O., 314, 321, *325*
Ostriker, J. P., 547, *555*
Ota, S., 243, *245*
Otter, M., 563, *572*
Ouimet, J. A., 1167, *1168*
Oulasvirta, A., 52, 53, *60*, 299, *312*
Ousterhout, J. K., 1031, *1039*
Ovaska, S., 1143, *1148*
Overbeeke, K., 17, *22*, 194, *196*
Övergaard, G., 993, *1016*
Overhage, J. M., 667, 677
Overhage, M., 667, *676*
Oviatt, S., 169, *175*, 395, *411*, 414, 415, 417, 418, 419, 420, 421, 422, 423, 424, 425, 428, *429*, *430*, *432*, 861, 863, *869*, 1254, *1265*
Oviatt, S. L., 181, *197*, 414, 415, 419, 420, 421, 422, 423, 424, 425, *430*, *431*, *432*, 861, 863, *869*
Owens, D., 218, *226*
Owens, J., 595, *598*
Ownby, D., 781, 782, *790*
Ownby, R., 786, 787, *790*
Oyama, S., 549, *556*
Oyler, M. A., 1167, *1168*
Ozmultu, H., 566, *572*
Ozok, A. A., 648, 651, *656*, *658*, 1153, 1158, 1163, 1166, *1169*
Ozyildiz, E., 415, *431*

P

Pachella, R. G., 46, *60*
Paciello, M. G., 864, *869*
Packer, H., 1181, *1188*
Padden, C. A., 886, 887, 888, *892*, *893*
Padmanabhan, V., 172, *172*
Paelke, G., 711*t*, 717
Paepcke, A., 171, *173*
Paganelli, L., 1123, *1125*
Pagani, D., 1027, *1039*
Page, L., 567, *571*
Pagulayan, R. J., 632, *638*, 748, 749, 750, 754, *758*, *759*, 1138, 1141, 1145, 1146, *1148*
Paier, D., 611, *618*
Pain, H., 839, 845, 848, *852*
Paine, G., 723, *737*
Paivio, A., 69, *74*, *75*, 492, *506*
Pak, A., 707, *718*
Palamara, P., 705, *718*
Palen, L., 546, 551, *556*, *557*, 642, *657*, 1242, 1249, 1253, *1265*
Paler, E. A., 723, *739*
Paley, M., 732, *738*
Palfreyman, K., 289, *292*, *293*
Palmer, E., 735, *739*
Palmer, J. C., 54, *60*
Palmer, J. W., 652, *657*
Pan, S., 439, *457*
Pan, Z., 84, *91*
Panayi, M., 835, 840, *852*
Pancer, S. M., 138, *146*
Pangaro, G., 194, *197*, 473, 475, 480, 484, *486*
Pankanti, S., 414, 417, *431*, 642, *658*
Panksepp, J., 79, *91*
Pao, C., 414, *432*
Pao, H., 914, *915*
Papert, S., 396, *411*, 477, *486*, 498, *506*, 614, *619*, 686, *700*, 797, 804, 806, *808*
Papin, J.-P., 1200, 1206
Pappas, C. J., 854, 858, *868*, *869*, 1254, *1263*
Paradiso, J., 194, *197*
Paramythis, A., 1085, *1105*
Parasuraman, R., 45, 47, 57, *60*, 120, 126, *131*, 574, 576, 582, 589, 595, *597*, *599*, *600*, 727, 728, 730, 732, 735, *739*, *740*, 1222, *1238*

Paris, C., 444, *455*
Parise, S., 139, 140, *146*
Park, D. C., 785, 786, *791*
Park, K. S., 554, *557*
Park, M., 223, 224, *226*
Parker, A., 972, *986*
Parker, D., *1265*
Parker, H., 220, *226*
Parker, L., 1250, 1257, *1263*
Parker, R. E., 818, *828*
Parkes, A. J., 473, *487*
Parkes, A. M., 702, *718*
Parra, L., 1227, *1239*
Parrish, R. K. I., 858, *869*
Parrott, G. W., 81, *91*
Parsons, J. E., 771, *774*
Parsons, M. W., 50, *60*
Pasanen, L., 259, *264*
Pashler, H., 35, *40*, 56, 57, *60*, 576, *600*
Passig, D., 772, *774*
Pasupathi, M., 706, *716*
Patel, A., 568, *572*
Patel, K., 874, *882*
Patel, M., 1143, *1148*
Patel, N. R., 668, 674, *677*
Patel, V. L., 668, 674, *677*
Patera, A., 474, *485*
Paternò, F., 864, *869*, 1108, 1111, 1123, *1124*, *1125*
Pathman, D. E., 674, *677*
Patil, S., 1253, *1265*
Patmore, D. W., 837, 841, *851*
Patrey, J., 300, *311*, 633, *637*
Patrick, A., 653, *657*
Patrick, K., 665, 670, *676*, *677*
Patten, J., 194, *197*, 473, 475, 478, *486*
Patterson, J. F., 289, *293*
Patterson, J. T., 38, *40*
Patterson, M., 548, *556*
Patterson, R. D., 250, *264*
Patton, J. W., 1069, 1070, 1074, *1080*
Patz, R. J., 99, *111*
Paulk, C. H., 723, *739*
Paulsen, C. A., 1143, *1148*
Pausch, R., 162, 167, *174*, 180, *197*, 204, *226*, 275, *293*, 419, *430*, 1123, *1125*
Pauzie, A., 711*t*, *718*
Pavel, M., 1228, *1238*
Pavlovic, V. I., 415, 417, 426, 427, *431*, *432*
Payne, C., 1056, *1059*
Payne, D. G., 724, *738*
Payne, J. W., 669, 673, *677*
Payne, S. J., 38, *40*, 50, *60*, 64, 65, 66, 67, 68, 69, 71, 72, *74*, *75*, *76*
Payne, S. L., 754, *759*
Payton, F. C., 642, *655*
Pazzani, M. J., 441, 449, 451, *455*, *457*
Pea, R., 804, *808*
Peacock, A., 650, *657*
Peacock, B., 702, *718*
Pearl, J., 596, *600*, *601*
Pecjak, V., 84, *91*
Peck, J., 1071, *1079*
Pedell, S., 1067, 1068, *1080*
Peden, M., 703, *718*
Pedersen, E., 550, *555*
Pedersen, J., 1066, 1067, 1071, 1072, 1074, *1080*
Peebles, D., 109, *112*
Pejtersen, A., 574, *600*
Pelachaud, C., 414, *428*
Pelican, M., 574, *600*
Pellegrino, B., 1119, *1124*
Pellican, S., 913, 914, *915*
Pelto, P. J., 1242, *1265*
Pelz, D., 498, *506*
Pena, P., 705, *718*
Penley, J. A., 121, *131*
Pennathur, A., 574, *600*
Pennington, C. A., 818, 837, *828*, 840, 845, *851*, 862, *869*
Pennington, N., 578, *600*
Pennock, D. M., 564, *572*
Pentland, A. P., 85, *90*, 170, *175*, 178, 180, 181, 185, *196*, *197*, *198*, 451, *457*, 862, *869*
Pentland, S., 417, 428, *429*, *431*
Pepe, A., 117, 122, *131*, 1145, *1148*
Percival, L. C., 45, *60*
Perdue, P. C., 1243, *1265*
Pereira, F. C. N., 415, *429*
Perelmouter, J., 841, 849, *851*
Peres, M., 1234, *1238*
Peretti, P. O., 84, *91*
Perez, M. A., 106, *111*
Pérez-Quiñones, M. A., 1142, 1143, *1149*
Perfetti, C. A., 69, *76*
Perkins, R., *987*, 1021, *1039*, 1137, *1149*
Perlman, G., 55, *60*, *378*
Perret, V., 1123, *1124*
Perrif, A., 647, *656*
Perrolle, J., 1257, *1261*
Perrott, D., 249, 252, *264*
Perry, J. F., 1226, *1239*, 1244, *1265*
Perry, M., 224, *225*
Perry, Z. W., 770, *775*
Person, L. H., Jr., 725, *737*
Persson, B., 220, *227*
Persson, P., 1256, *1265*
Pervin, E., et al., 1035, 1039
Pesanti, L. H., 547, *556*
Petajan, E. D., 417, *429*, *431*
Peters, N., 596, *598*
Peterson M. W., 875, *882*
Peterson, B., 628, *636*
Peterson, L. R., 52, *60*
Peterson, M. J., 52, *60*
Peterson, R., 417, *429*, 579, *602*, 840, *850*
Petre, M., 502, *506*, 1176, *1188*
Petrelli, D., 452, *457*
Petrie, H., 863, *869*
Petroski, H., 579, *600*
Petrosky, H., 1194, *1207*
Pettersson, J. H., 644, *657*
Pettey, M., 169, *174*
Petty, R. E., 141, *146*
Pew, R., 4, 5, 6, 8, *24*
Pew, R. W., 109, *112*, 1195, 1196, *1207*
Pfendler, C., 217, 218, *227*
Pfleger, N., 415, *431*
Pham, H., 190, *198*
Phanariou, C., 1123, *1124*
Phelps, E. A., 78*n*, *90*

Phelps-Goodman, E., 475, *486*
Phillips, D. J., 1253, *1265*
Phillips, J., 415, *432*
Phillips, L. H., 495, *506*
Phillips, P., 270, 277, *293*
Phillips, R. J., 185, *199*
Piaget, J., 684*t*, *700*, 794, *808*
Pianesi, F., 454, *456*
Piazzalunga, U., 650, *657*
Picard, A., 82, 85, 87, 88, *90*
Picard, R. W., 78, 82, 85, 86, 87, 88, *90*, *91*, 178, 182, *197*, *198*, 439, 451, *456*, *457*
Picardi, M. C., 55, *61*
Piccoli, B., 317, *325*
Pickering, A., 968, *987*
Pickering, J., 181, *198*
Pickover, C., 496, *506*
Pidcock, F. S., 833, *851*
Pidermann, M., 316, *325*
Piechulla, W., 706, *718*
Pieke, R., 169, *173*
Pieper, M., 845, *851*
Pier, K., 169, *173*, 550, *555*, 1033, *1038*
Pierard, C., 1234, *1238*
Pierce, J., 165, 171, *174*, 180, *197*
Pierce, T., 361, *379*
Piernot, P., 474, *486*
Pilar da Silva, D., 864, *869*
Pilcher, J. J., 117, *131*
Pillet, B., 646, *656*
Pine, B. J., 904, *915*
Pine, J., 976, *987*
Pinhanez, C., 192, *196*
Pinkard, N., 764, *774*
Pinkerton, M., 192, *196*
Pinski, G., 567, *572*
Pinsonneault, A., 546, 550, *556*
Piper, A. M., 138, *146*
Piper, B., 475, 481, 484, *486*, *487*
Pique, M., 163, 167, *173*, *175*
Pirkl, J. J., 814, *828*
Pirolli, P., 108, *111*, *112*, 540, *543*
Pister, K. S. J., 179, *197*
Pitette, G., 1121, *1124*
Pittman, J. A., 414, 427, *429*, *430*
Plainis, S., 218, 219, *227*
Plaisant, C., 163, *176*, 181, *198*, 797, 798, 799, *807*
Plaisted, D., 299, *312*
Platner, M., 797, 799, *806*, *807*
Platt, J. C., , 186, *198*, 435, *457*
Platz, A., 684, *699*
Playfair, W., 517, *543*
Pleasant, A., 878, *883*
Pleydell-Pearce, K., 1230, *1239*
Plomp, T., 765, *774*
Plous, S., 576, 578, 595, *600*
Plowman, T., 974, *987*
Plude, D., 781, *792*
Plutchik, R., 87, *91*
Poddar, I., 415, *431*
Poehlmann, K. M., 85, *89*
Poggio, T., 187, *196*
Pohl, K., 1046, *1059*
Pohl, W., 1090, *1105*
Pohlmann, K., 252, *264*
Polen, L., 22, *24*
Polese, G., 840, *849*, 862, *866*
Polifroni, J., 414, *432*
Polk, T. A., 100, *112*
Pollack, I., 415, *432*
Pollack, M. E., 436, *456*
Polson, P. G., 56, *58*, 95, 98, 99, 102, 103, *111*, *112*, 502, *505*, 1047, 1052, *1059*, 1175, 1176, *1189*, *1190*, 1205, *1207*
Polt, J. M., 1223, *1238*
Poltrock, S. E., 531, *542*, 1248, *1265*
Ponce, J., 179, *196*
Ponnekanti, S., 1123, *1125*
Poole, M. S., 547, *555*
Poon, L. W., 788, *790*
Pope, A. T., 1227, *1239*
Pope, S. T, 1033, *1038*
Popescu, G. V., 626, *637*
Porcari, J. P., 139, *146*
Port, S. C., 183, *197*
Porteous, J., 797, *806*
Porter, J. M., 563, *571*
Porter, L. W., 746, *759*
Porter, W. L., 1072, *1079*
Poschen, M., 1065, 1069, *1081*
Posner, M. I., 28, 29, 31, 32, 33, *39*, *40*, 44, 55, 56, *59*, *60*
Possamäi, C.-A., 51, *59*
Postma-Schuit F. C. E., 669, *677*
Poston, R., 232, *245*
Postrel, V., 902, *915*
Potamianos, G., 414, 415, 417, 424, 427, 428, *430*, *431*
Poteat, G. M., 973, *987*
Potosnak, K., 167, *175*
Potter, E. E., 780, *791*, 1255, *1263*
Potts, A. L., 668, *677*
Potts, C., 644, *656*, 1054, *1057*, *1059*
Poulton, E. C., 578, 595, *600*
Pountain, D., 625, *637*
Poupyrev, I., 162, 168, *173*, *175*, 182, *197*
Povel, D. J. L., 491, *507*
Povenmire, H. K., 632, *637*
Power, D. J., 668, *678*, 886, *893*
Power, M., 845, *852*
Powledge, P., 1253, *1262*
Praag, J. V., 788, *791*
Prabhu, P., 574, *598*
Prakash, A., 552, *557*
Pratt, A., 876, *883*
Pratt, J. D., 31, 32, 33, 34, 35, *39*, *40*, *41*, 54, *62*
Precht, M., 211, 212, 213, *227*
Preece, J., 6, *24*, 574, 575, *600*, 604, 605, 606, 609, 610, 611, 617, *619*, 742, *759*, 781, *791*, 824, *828*
Preis, A., 250, *263*
Presser, S., 1153, *1168*
Presti, S. L., 652, *657*
Prevas, K., 108, *111*, 1201, *1207*
Prevett, T., 732, 735, *740*
Prevost, S., 78, *89*, 417, *429*
Prevot, T., 735, *739*
Pribram, K. H., 575, *600*
Price, K. J., 384, *391*, 835, *851*
Price, M. N., 172, *176*
Price, S., 795, *808*
Prietula, M. J., 141, *145*, *146*, 576, *599*

Prinz, W., 50, *59*, 551, *557*
Prinzel, L. J., 1227, *1238*, *1239*
Pritchett, A. R., 593, *600*, 732, 734, *739*
Priyantha, N., 180, *198*
Proctor, R. W., 28, 29, 36, 38, *40*, *41*, 44, 48, 49, 50, 54, 57, *59*, *60*, *61*
Proffitt, D., 167, *174*, 180, *197*, 1222, *1239*
Proft, D., 795, *807*
Propp, V., 1044, *1059*
Prothero, J. D., 630, *636*
Provenzo, E. F., 742, *759*
Pruitt, J., 18, *24*, 921, *926*, 941, 978, 979, 980, *987*, 993, *1015*, *1016*, 1131, *1149*
Pryor, J., 139, *146*
Psihoios, J., 218, *227*
Psillos, S., 593, *600*
Puerta, A., 1123, *1125*
Pulat, B. M., 724, *739*
Purao, S., 1055, *1058*
Purves, B., 879, *882*, 1077, *1078*, *1080*
Putnam, R. D., 548, *557*, 1254, *1265*
Putz-Anderson, V., 168, *175*
Pylyshyn, Z., 730, *739*

Q

Qin, Y., 575, *597*
Qu, N., 764, *774*
Quatrano, L. A., 837, *850*
Quesenbery, S., 1134, *1149*
Quimby, A., 711*t*, *718*
Quin, Y., 106, *111*
Quinn, A., 298, *312*
Quinn, D. M., 770, *774*
Quintana, C., 879, *808*, *882*
Quintanar, L., 139, *146*

R

Rabbitt, P. M. A., 31, *40*, 814, *828*
Rabiner, L. R., 181, 187, *198*, 383, *391*
Radvansky, G. A., 68, *76*, 575, 576, *600*
Radwin, R. G., 843, 847, 848, *851*, *852*
Rafaeli, Y., 903, *915*
Rafal, R. D., 31, 33, *39*, *40*
Raffle, H. S., 473, 474, 477, *487*
Rafii, A., 181, *198*
Raghavan, S., 914, *915*
Raghavendra, P., 825, *827*
Rahardja, K., 87, *91*
Rahman, T., 224, *227*
Raiello, P., 1141, *1147*
Rainie, L., 604, *619*
Raisinghani, M. S., 583, *599*
Raj, A. K., 1226, *1238*, *1239*
Rajasekaran, M., 611, *619*
Rakotonirainy, A., 643, *657*
Ralston, J., 610, *618*
Ramduny, D., 269, 274, 289, *292*, *293*
Ramey, J., 909, *916*, 934, 935, *947*, 1063, *1081*, 1133, 1140, *1147*
Ramloll, R., 862, *870*
Ramos, G., 165, 171, *174*
Rampp, D. L., 862, *869*
Ramsey, C., 887, *893*
Ramstein, C., 862, *869*
Ramzy, C., 673, *676*
Randall, D., 729, *738*, 908, *915*, 967, *986*, *987*
Rankin-Erickson, J., 845, *851*
Ransby, M., 876, *882*
Rao, R., 539, *543*, 548, *556*
Rao, S. M., 50, *60*
Raphael, C., 767, *774*
Rapson, R. L., 83, *90*
Raskar, R., 171, *175*
Raskin, J., 166, 167, *175*, 1018, *1039*
Rasmussen, C., 552, *557*
Rasmussen, J., 123, 125, *131*, *132*, 574, *576*, 577, 581, 583, *600*, *602*, 723, *740*
Ratcliff, R., 48, *61*
Rathje, W. L., 970, 974, *987*
Ratti, C., 475, *486*, *487*
Ratzan, S. C., 665, *676*
Ratzer, A., 1030, *1039*
Rau, M. T., 820, *828*
Rauterberg, M., 475, *487*, 502, *507*, 862, *867*
Raven, M. E., 549, *556*
Ravenell, K., 873, *883*
Ravishankar, M., 308, *312*
Rawls, J., 1245, *1265*
Raybourn, E. M., 797, *809*
Rayport, J. F., 977, *987*
Rayson, P., 273, *293*
Rayward-Smith, V., 580, *600*
Reader, W. R., 69, *76*
Reagle, J., 643, 647, *655*
Reason, J., 148, 149, *157*, 576, 579, *600*
Recht, B., 475, *486*
Reddig, C., 414, 430
Reddy, R., 308, *312*
Redelmeier, D. A., 705, *718*
Redish, G., 1130, *1147*
Redish, J. C., 751, *758*, 922, *926*, 928, 930, 934, 935, 936, 940, 943, *947*, 1132, 1133, 1136, 1139, 1140, 1143, *1147*, 1276, *1277*
Redmiles, D., 646, *656*
Redondo, M., 50, *61*
Redstr, J., 474, *486*
Ree, M. J., 631, *635*
Reed, E. S., 123, *131*, 138, *146*
Reed, J., 804, *808*
Reed, S. E., 1065, *1080*, 1119, *1124*
Reeder, R. W., 108, *111*, 648, *657*
Reeve, T. G., 50, *61*
Reeves, B., 79, 81, 82, 84, 87, *89*, *91*, 137, 139, 140, *146*, 403, 404, *411*, 570, *572*, 859, *869*, 901, *915*, 1231*n*, 1253, 1256, *1265*
Reeves, C., 580, *600*
Reeves, L. M., 626, 630, *638*, 1222, 1223*n*, 1236, *1239*
Regal, D. M., 733, *739*
Reicher, G. M., 496, *507*
Reid, F. J. M., 1065, *1080*
Reid, S. R., 723, *738*
Reidl, M. O. 1201, *1207*
Reilly, D., 308, *312*
Reilly, J., 442, *457*
Reilly, W. S., 83, *89*
Reimann, R., 993, *1015*, 1054, *1058*
Rein, G. L., 15, *22*, 550, 555
Reinen, I. J., 765, *774*

Reinfurt, D. W., 704, *718*
Reis, J., 140, *146*
Reisberg, D., 81, *90*
Reiser, C., 764, *775*
Reiser, R. A., 605, *619*
Reisin, F.-M., 1062, *1078*, 1250, *1262*
Reising, D., 577, *600*
Reisner, P., 502, *507*
Reiswig, G., 862, *868*
Reiter, M., 643, 649, *656*, *657*
Reithinger, N., 414, 430, 431
Reitsma, P., 800, *807*
Rejmer, P., 1254, *1262*
Rekimoto, J., 171, *175*, 181, 182, *197*, *198*, 474, *487*, 1112, *1125*
Rekola, S., 943, *947*
Remington, R. W., 31, *39*, 1198, *1207*
Remondeau, C., 414, *432*
Rempel, D., 168, *174*, *175*, *176*
Remus, W., 15, *24*
Ren, J., 646, *656*
Ren, S., 475, *486*
Renaud, K., 648, 649, *656*, *657*
Rennels, G., 583, *598*
Renshaw, E., 435, *457*
Rentsch, J. R., 74, *75*
Repenning, A., 804, *808*
Resner, B., 475, *486*
Resnick, M., 474, 485, *487*, 804, *808*
Resnick, P., 267, *293*, 548, 553, *557*, 796, 805, 806, 1251, *1262*
Resnikoff, H. L., 515, 517*t*, *543*
Rettig, M., 1025, *1039*, 1075, *1080*
Reuscher, T. R., 706, *717*
Reuter-Lorenz, P., 85, *91*
Reuth, R., 223, *227*
Revelle, G., 799, *807*
Revelle, W., 128, *131*, 901, *915*
Rex, D. B., 541, *543*
Rey, G., 1111, *1125*
Reynard, G., 554, *555*
Reynolds, C., 178, *198*
Rheingens, P., 651, *656*
Rheingold, H., 604, *619*, 974, *987*
Rhoades, D. G., 1176, *1189*
Rhoades, J. A., 547, *555*
Rhodes, B. J., 442*n*, *457*, 547, *557*
Rhodes, E., 674, *677*
Rhodes, J. S., 1276, *1277*
Rhyne, J., 172, *176*
Ribbens, W. B., 702, *718*
Riccardi, G., 440, *457*
Rice, D. P., 663, *676*
Rice, R. E., 552, *555*, 610, *619*
Rich, C., 437, 438, *456*, *457*
Rich, M., 970, *987*
Richard, C. M., 3*341*
Richards, B., 1077, *1081*
Richards, J. T., 84, *91*, 504, *506*, *507*
Richardson, B., 702, *718*
Richardson, C., 169, *173*
Richmond, M. J., 623, *636*
Richter, L., 121, *130*
Richter, M. M., 644, *655*
Ricker, K. L., 33, 35, *39*, *40*
Riedl, J. T., 519, *542*, 553, *555*, *556*, *557*
Riedl, M. O., 110, *112*, 1201, *1207*
Riegelsberger, J., 583, *601*, 1254, *1265*
Rieh, S. Y., 142, *146*
Rieman, J., 95, 101, *112*, 570, *572*, 1047, *1059*, 1175, 1176, *1189*, *1190*
Riesbeck, C. K., 581, *601*
Riggio, L., 33, *40*
Righi, C., 908, *915*
Riley, V., 576, 582, 595, *600*, 727, 730, 732, *739*
Rime, B., 170, *175*
Ringel, M., 194, *198*
Risch, J. S., 541, *543*
Risden, K., 796, *807*, 1143, *1148*
Ritter, F. E., 94, 108, 110, *112*, *113*, 575, *601*
Ritterfeld, L., 683, *700*
Riva, G., 138, 139, *145*, *146*
Rivera, K., 88, *91*
Riviere, C. N., 784, *791*
Rizzo, A. A., 634, *637*
Rizzolatti, G., 33, *40*
Rizzolo, M. A., 670, *677*
Ro, O., 218, *225*
Roads, C., 253, 254, *264*
Robbins, D., 172, *173*
Robbins, M. A., 786, *790*
Robert-Ribes, J., 415, 424, *431*
Roberts, J. M., 609, *618*
Roberts, S. D., 139, *144*
Roberts, T., 17, *22*, 472, *486*, 1130, *1149*
Robertson, G. G., 162, *173*, 414, 417, *432*, 517*t*, 518, 538, 541, *542*, *543*, 566, 567, *571*, *572*, 859, 861, *870*
Robertson, G. R., 406, *411*
Robertson, T. S., 141, *145*, 1066, *1080*
Robin, H., 516, 527, *543*
Robins, J., 1066, *1080*
Robinson, C., 935, *947*, 1069, 1070, 1074, *1080*
Robinson, E., 549, *558*
Robinson, J. E. III, 563, *572*, 735, *739*
Robinson, P., 837, 839, 847, *850*
Robinson, R. E., 966, 977, *987*
Robinson, T. N., 665, 666, 670, 671, *677*, 781, *791*
Robinson-Staveley, K., 769, *774*
Robin, H., 516, 518*f*, 527, *543*
Robson, G., 892, *893*
Rocco, E., 553, *557*, 1254, *1265*
Rockwell, T., 578, *601*
Rodden, T., 274, 277, 283, 289, *292*, *293*, 841, *849*, 967, *915*, *986*, *987*
Rode, J., 646, *656*
Rodgers, M. D., 734, *738*
Rodgman, E. A., 704, *718*
Rodvold, M., 574, *601*
Roe, D. B., 383, *391*
Roediger, H. L., III, 54, *61*
Roessler, A., 138, *145*
Roettling, M., 122, *131*
Rogers, D., 795, *807*
Rogers, G. T., 1123, *1125*
Rogers, W. H., 734, *739*, *790*, *791*
Rogers, W. A., 778, 784, 786, 787, *791*
Rogers, Y., 394, 396, *411*, 574, 575, 576, *600*, *601*, 604, 609, *619*, 742, *759*, 795, 798, 799, *808*, 986*n*, *987*
Rogozan, A., 417, *431*
Rohall, S. L., 435, *456*
Rohn, J. A., 1130, *1149*, 1172, *1189*
Rohs, M., 270, *294*

Rolland, J. P., 224, *225*, 630, *637*
Romero, R. L., 748, 751, 753*t*, *759*, 1138, 1146, *1148*
Rommelse, K., 172, *174*, 191, *197*, 453, *456*
Romney, A. K., 969*n*, 972, *987*
Ronkainen, S., 259, *264*
Rönkkö, K., 1015, *1016*
Roos, J., 581, *600*
Root, J., 873, *882*
Root, R. A., 1046, *1059*
Root, R. W., 552, *555*, 754, *759*, 1069, *1080*, *1081*
Ropper, A., 833, *851*
Rorty, R., 1246, *1265*
Rosa, R. H., Jr., 854, *868*, 1254, *1263*
Rosch, E., 1043, *1059*
Roscoe, S. N., 3, *24*, 44, *61*, 632, *637*, *638*, 723, 726, 735, *737*, *739*
Rose, A., 1251, *1265*
Rose, C. F., 420, *430*
Rose, K., 804, *807*
Roseman, I. J., 83, *91*
Roseman, M., 1033, *1039*
Rosen, B., 786, *791*
Rosen, L. D., 764, *775*
Rosenbaum, H., 1248, *1264*, *1265*
Rosenbaum, S., 909, *916*, 1130, *1149*, 1172, *1189*
Rosenberg, B., 182, *196*
Rosenberg, R., 859, *869*
Rosenberg, S., 1256, *1265*
Rosenblitt, D. A., 548, *556*
Rosenbloom, P. S., 56, *60*, 503, *506*, 575, *599*
Rosenfeld, A., 85, *91*, 185, *199*
Rosengren, K. E., 88, *91*
Rosenholtz, R., 171, *172*
Rosenstock, I. M., 138, *146*
Rosenthal, M., 832, *851*
Rosenthal, R., 766, 769, *774*
Rosenzweig, A. S., 81, *90*
Roshan, S., 980, *987*
Rosien, A., 109, *111*
Roskam, A. J., 708, *718*
Roskos, E., 230, *244*
Ross, A., 417, 428, *430*
Ross, J. M., 117, 118, *131*
Ross, L., 126, 127, *146*
Ross, T., 711*t*, 718
Rosson, M. B., 1032, *1039*, 1043, 1044, 1045, 1046, 1047, 1049, 1052, 1053, 1054, 1055, 1056, *1057*, *1058*, *1059*, 1108, 1116, *1125*
Rostom, A., 669, *677*
Rotenberg, M., 1247, *1261*, *1265*
Roth, A. E., 1253, *1265*
Roth, D., 782, *790*
Roth, E., 574, 575, 576, 577, *597*, 582, *601*
Roth, S. F., 517, 536, *542*, *543*
Rothbaum, B. O., 634, *635*, *636*, *638*
Roto, V., 299, *312*
Rouet, J. F., 69, *76*
Roufs, J., 219, *225*
Rougui, J., 1121, *1124*
Roulland, F., 501, *506*
Rouncefield, M., 583, *597*
Rourke, L., 610, *618*
Rouse, W. B., 581, 583, 591, *597*, *601*
Rousseau, G. K., 787, *791*
Roussel, N., 277, *294*, 1033, *1038*
Rovner, D., 669, *677*
Rowberg, A. H., 935, *947*
Rowland, C., 874, *882*
Rowley, D. E., 1176, *1189*
Rowley, P., 163, *173*
Roy, D. M., 428, *429*, 840, *852*, 862, *869*
Roy, E. A., 35, *41*
Royer, C., 109, *111*
Ruben, H., 496, *507*
Rubert, M. P., 782, *790*
Rubichi, S., 36, *40*, 50, *61*
Rubin, A. M., 88, *91*, 643, *657*
Rubin, B., 492, *506*
Rubin, D. C., 501, *507*
Rubin, J., 800, *808*, 1130, 1132, 1134, 1136, 1139, 1140, *1149*
Rubin, K. S., 575, *601*, 1055, *1059*
Rubin, P., 415, 426, *431*
Rubine, D., 185, *198*
Rubinson, H., 580, *599*
Rubinstein, R., 924, *926*
Rudd, J., 1135, *1149*
Rudd, R. E., 873, *882*
Rudell, A. P., 36, *41*
Rudisill, M., 51, *59*
Rudmann, S., 593, *598*, *601*
Rudnicky, A., 389, *391*, 419, *431*
Rugg, M. D., 47, *61*, 125, *131*
Ruhleder, K., 548, *557*
Rui, Y., 180, *198*
Rumbaugh, J., 1055, *1058*
Rupert, A. H., 1226, *1239*
Rusk, N., 804, 806, *808*
Russell R., 946, *947*
Russell, D. M., 540, *543*
Russell, J. A., 79, 87, *89*, *90*
Russell, M. J., 417, *432*, 1139, *1149*
Russell, S., 580, *601*
Russo, R., 46, *59*
Rutgers, E. J. T., 669, *677*
Ruthruff, E., 576, *600*
Ruthsatz, J. M., 813, *827*
Ruthven, I., 566, *572*
Rutkowski, C., 36, *40*
Rutledge, J., 163, 167, *175*
Rutter, J., 744, *759*
Ryall, K., 171, *176*, 474, *485*
Ryan, R. M., 122, 123, 129, *130*
Ryan, R. R., 117, 122, *132*
Ryder, J., 1197, *1208*
Ryder, J. W., 862, *869*
Ryokai, K., 475, *487*
Ryu, Y. S., 360, *378*

S

Saaksjarvi, M., 1077, *1080*
Saariluoma, P., 52, *60*
Saarinen, T., 1077, *1080*
Sabeh, R., 724, *739*
Saberi, K., 249, *264*
Sacerdoti, E. D., 581, *601*
Sachs, P., 966, *987*
Sacks, W., 785, *790*
Sadosky, T. L., 723, *738*

Sadowski, W., 630, 631, *638*
Sadralobadi, T., 249, *264*
Saffer, D., 903, *915*
Said, E., 968, *987*
Sainfort, F., 667, 671, 672, *676*, *677*, *678*, 861, *868*, *869*
Sainfort, P. C., 315, 323, *326*
Saint, S. P., 736, *739*
Saito, S., 317, 318, 319, *325*
Saito, T., 863, *868*
Sajda, P., 1227, *1239*
Saka, C., 420, 422, *432*
Sakai, J., 835, 843, 847, *851*
Salant, P., 972*n*, 987
Salas, E., 72, *75*, 121, *130*, *132*, 574, 583, *597*, *600*, *601*, 1195, *1207*
Saldarini, R., 1087, *1105*
Salem, C., 843, 849, *852*
Salesin, D., 87, *90*
Salingaros, N. A., 491, *506*
Salisbury, K., 231, *245*
Salthouse, T. A., 46, *61*, 96, *112*, 814, *828*
Salton, G., 566, *572*
Saltzer, J., 646, 648, *657*
Salvador, T., 1068, 1071, *1080*
Salvaneschi, P., 650, *657*
Salvendy, G., 361, *377*, 628, 631, *638*, 648, *658*, 859, *870*, 1153, 1158, 1163, *1169*
Salvucci, D. D., 107, 108, *111*, *112*, 582, *599*, 1172, *1189*, 1201, *1207*
Salzman, M., 1179, 1180, 1186, 1187, *1188*, 1249, *1265*
Samman, S., 626, *638*
Sammet, J., 3, *24*
Samuel, J., 1054, *1058*
Sanborn, S., 574, *598*
Sanchez, M. E., 1167, *1169*
Sanders, A. F., 46, 48, 49*f*, *61*
Sanders, E. B.-N., 1067, 1068, 1069, 1070, 1074, 1075, *1080*
Sanders, M. S., 207, 209, *227*, 248, *264*, 723, 724, *739*, 1228, *1239*
Sanders, P., 634, *638*
Sanders, T., 736, *740*
Sanderson, P., 577, *600*
Sandin, D. J., 623, *636*
Sandor, O., 277, *294*
Sandry, D. L., 395, 411, 420, *432*
Sanford, B. D., 735, *739*, *740*
Sangha, A., 279, *294*
Sano, D., 394, 403, 404, 409, *411*
Sanocki, E., 549, 555
Santarelli, T., 1197, *1208*
Santoro, T. P., 1201, 1204, 1205, *1207*
Saracevic, T., 565, *572*
Sarasohn-Kahn, J., 873, *881*
Sarge, A., 84, *90*
Sarin, R., 172, *173*, *174*, 439, *456*
Sarin, S. K., 551, *554*
Sarter, N. B., 582, *601*, 734, *739*
Sarvas, R., 270, *294*
Sarwar, B., 642, *656*
Sas, C., 449, *455*
Sasse, M. A., 583, *601*, 642, 647, 654, *655*, *657*, 1254, *1265*
Satabin, P., 1234, *1238*
Satava, R., 634, *638*
Sato, S., 171, *176*, 1069, 1071, *1080*
Satran, A., 394, 404, *411*
Satyanarayanan, M., 287, *293*
Satzinger, J., 547, *557*
Saund, E., 166, 169, *175*, *176*
Saunders, T., 121, *132*
Sauro, J., 1138, *1149*
Savant, G., 204, *225*
Savidis, A., 1085, 1098, 1100, 1101, 1102, *1105*, 1111, *1125*
Sawant, S., 874, *882*
Sawhney, M., 901, 902, *915*
Sawhney, N., 169, *176*, 262, *264*
Sawyer, J. E., 583, *598*
Sawyer, P., 908, *915*, 966, *986*, 1184, 1186, *1189*
Sawyer, S., 1248, *1265*
Scaife, M., 394, 396, *411*, 576, *601*, 795, 798, 799, *808*
Scallen, S. F., 729, *739*
Scane, R., 224, *225*
Scanlon, D., *876*, *883*
Scanlon, T., 403, *411*
Scapin, D. L., 1176, 1180, *1188*, *1189*
Scardamalia, M., 805, *808*
Scargle, S. D., 841, *849*
Scerbo, M. W., 120, 121, 122, *132*, 582, *598*, *600*, 1227, *1238*
Scha, R., 521, *542*
Schaab, J. A., 843, *852*
Schaaf, R., 634, *635*
Schachtel, E. J., 84, *91*
Schachter, S., 85, *91*
Schadt, M., 213, *227*
Schaeken, W., 576, *601*
Schafer, J. B., 442, *457*, 553, *557*, 642, *656*
Schaffer, M. M., 1043, *1059*
Schaie, K. W., 778, *790*
Schank, R. C., 581, *601*, 818, *828*, 862, *869*
Schapire, R. E., 186, *198*
Scharaffa, A., 219, *227*
Schatz, B. R., 547, *557*
Schauer, H., 841, 849, 851, *867*
Scheckler, W., 674, *677*
Scheffler, S., 1245, *1265*
Scheidt, P., 874, *882*
Scheier, M. F., 116, 119*f*, 121, *130*, *132*
Schein, O. D., 858, *869*
Schell, R., 647, *655*
Schenkmann, B., 220, *227*
Scherer, K. R., 83, 86, 87, *89*, *91*, 116, 117, *132*
Scherer, M. J., 846, 848, *852*
Scherlis, W., 548, *556*
Scheutz, M., 83, *91*
Schiano, D. J., 85, 87, *90*, *91*, 549, 550, *556*, *557*, 1256, *1265*
Schiaratura, L., 170, *175*
Schieber, F. J., 784, *791*, 854, 858, *869*
Schiele, B., 451, *457*
Schiff, L. R., 1249, *1266*
Schiffman, J. C., 858, *869*
Schilit, B. N., 171, 172, *176*
Schindhelm, R., 710, *717*, *718*
Schkade, D. A., 580, *599*, 673, *678*
Schlag, C., 842, *850*
Schlegel, R. E., 1223, 1225*t*, *1239*
Schleyer, T. K. L., 1167, *1169*
Schlick, C., 217, 218, 223, *227*
Schlosser, A. E., 139, *146*, 547, *555*
Schmandt, C. M., 169, *176*, 262, *264*, 278, *294*, 389, *391*, 548, *557*
Schmauks, D., 414, *430*
Schmidt, A., 182, 191, *196*, *198*, 1036, *1038*
Schmidt, G., 1062, *1078*, 1250, *1262*

Schmidt, H. P., 736, *739*
Schmidt, K., 736, *739*, 966, *987*
Schmidt, L., 122, *131*, 218, 219, 223, 224, *226*
Schmidt, R. A., 70, 71, 72, *75*, *76*
Schmitt, B. H., 904, *915*
Schmitz, J., 610, *619*
Schmorrow, D. D., 57, *61*, 300, *311*, 625, *638*, 1222, 1223, 1223*n*, 1231*n*, 1233, 1236, *1239*
Schneider, F. B., 1247, 1254, *1265*
Schneider, K. G., 794, 795, 796, 806, *808*
Schneider, L., 970, *987*, 1063, *1077*
Schneider, M., 439, *457*
Schneider, S. J., 139, *146*
Schneider, W. X., 31, 33, *39*, 119, 120, 125„ *132*
Schneiderman, B., 725, *739*
Schneier, B., 273, *294*, 646, 647, *657*, *658*
Schoelles, M. J., 109, *112*
Schoeman, F. D., 1253, *1265*
Schoen, C., 662, *678*
Schofield, J. W., 769, 770, *774*
Scholtz, J., 1131, *1149*
Scholz, J., 315, *324*
Schomaker, L., 414, *428*
Schön, D. A., 1044, *1059*, 1072, *1079*
Schoonen, R., 875, *883*
Schraefel, M. C., 441, 446, 450, *458*, 568, *572*
Schragen, J. M., 575, *601*
Schreiner, M. E., 804, *807*
Schroeder, M., 646, 648, *657*
Schroeder, R., 583, *601*
Schroeder, W., 403, 411, 1033, *1039*
Schroit, D., 17, *22*
Schuchard, R., 858, *867*
Schuchardt, K., 1074, *1078*
Schuhmann, D., 675, *677*
Schuler, D., 575, *601*, 797, 798, *808*, 982, *987*, 1062, 1063, *1081*
Schulert, A. J., 1123, *1125*
Schulmeister, R., 888, *892*
Schulten, K., *415*, 432
Schultz, E. E., 54, *61*
Schultz, R., 581, *600*
Schulze, K., 446, *458*
Schumacher, H. R. Jr., 832, *852*
Schumacher, P., 765, *774*
Schumann, C. E., 784, 785, *790*
Schumann, D. W., 138, 142, *146*
Schunn, C., 575, *599*, 582, *600*, *601*
Schvaneveldt, R. W., 57, 58, *61*
Schwaber, K., 912, *915*
Schwartz, J.-L., 415, *431*
Schwartz, L. K., 781, *792*
Schwartz, M. D., 674, *677*
Schwartz, N., 81, *91*
Schwartz, S. H., 365, *380*
Schwarz, A. J., 204, *225*
Schwarz, H., 549, *556*
Schwarz, N., 754, *759*
Schwarzkopf, E., 449, *456*
Schweickert, R., 52, 56, *61*
Schweitzer, J., 706, *716*
Schwoerer, C., 786, *791*
Scialfa, C. T., 858, *868*
Sciarrone, F., 442, *457*
Sciolto, T. K., 33, *40*
Scott, G. C., 668, *678*
Scott, I. U., 854, 854*f*, 858, 859, 861, *868*, *869*, 1254, 1255, *1263*
Scott, J. P., 179, *197*, 610, *619*, 972, *987*
Scott, N., 644, *655*, 1139, *1148*, 1244, *1265*
Scott, R., 574, *601*
Scott, S. D., 171, *176*, 194, *197*, 270, *294*
Scranton, P., 1243, *1265*
Scriven, M., 1046, *1059*
Scrivener, S. A. R., 1063, *1081*
Scurfield, R., 703, *718*
Seacord, R., 1119, *1124*
Seamon, A. G., 633, *636*
Seamon, J., 575, 576, *601*
Searle, A., 841, *850*, *851*
Searle, J. R., 1248, 1256, *1265*
Sears, A., 163, 167, *176*, 181, 190, *196*, *198*, 310, *311*, *312*, 384, *390*, *391*, 453, *457*, 675, *677*, 835, 838, 844, 847, *850*, *851*, *852*, 854, *868*, 1141, *1149*, 1176, 1177, 1178, 1179, 1182, 1185, *1189*
Sears, D. C., 764, 775
See, J. E., 45, *61*
See, K., 583, *599*
Seeger, C. M., 29, *39*, 50, *59*
Seemann, M., 675, *677*
Segal, R. B., 435, *457*
Segall, P., 1065, *1081*
Seger, D., 667, 668, *676*
Seidler, K., 724, *739*
Sekiyama, K., 424, *432*
Sekuler, A. B., 31, *40*
Seland, G., 1068, *1081*
Selcon, S. J., 734, *739*
Self, C. S., 141, *146*
Seligman, M. E. P., 122, *132*
Seligmann, D., 310, *311*
Selker, T., 162, 163, 167, *175*, *176*, 568, *572*
Sellen, A. J., 163, 167, 169, 172, *174*, *175*, *176*, 549, *555*, 643, 647, *655*, *657*
Selwitz, A., 1144, *1149*
Selye, H., 116, *132*
Semmlow J., 219, *226*
Sen, S., 442, *457*
Senders, J. W., 148, *157*
Seneff, S., 414, 415, *432*
Sengers, P., 1251, *1265*
Senior, A., 414, *430*, 642, *658*
Sequin, C., 167, *175*
Serafin, C., 711*t*, 716, 717
Serina, E., 168, *174*, *176*
Sethi, Y., 415, *431*
Setterlund, M. B., 81, *91*
Severson, R. L., 1251, 1253, *1262*, *1263*
Sewell, D. R., 580, *601*
Seymour, T., 110, *112*
Sfyrakis, M., 1085, *1105*
Shabot, M. M., 667, *676*
Shackel, B., 2, 3, 19, *24*, 673, *678*, 1109, *1125*
Shadmehr, R., 634, *635*
Shafer, G., 596, *601*
Shafer, S., 188, *196*, *198*
Shaffer, L. H., 50, *61*
Shahar, Y., 668, *677*
Shahmehri, N., 445, *455*, 1254, *1261*
Shaikh, A., 427, *432*
Shalin, V. L., 575, 581, *601*

Shallice, T., 396, *411*
Shamp, S., 138, *146*
Shankar, U., 646, *658*
Shankweiler, D., 887, *892*
Shannon, C. E., 8, *24*
Shapiro, D. C., 71, *76*, 729, *738*, *915*, 967, 986*n*, *986*, *987*
Shapiro, J. A., 446, *458*
Shapiro, S. C., 414, 415, 426, *430*
Shappell, S., 578, *602*
Sharda, R., 668, *678*
Sharit, J., 778, 780, 782, 784, 785, 786, 787, 788, *790*, *791*, *792*
Sharkey, P., 858, *869*
Sharkey, T. J., 629, *637*
Sharma, G., 579, *602*
Sharma, R., 415, 426, *431*, *432*
Sharp, H., 574, 575, *600*, 604, 609, *619*, 742, *759*
Sharp, L., 1242, *1265*
Sharp, R., 270, *294*
Shaw, C., 770, *775*
Shaw, E., 879, *882*
Shaw, M., 1121, *1124*
Shaw, R., 843, 849, *852*
Shawe-Taylor, J., 186, *196*
Shea, J. B., 70, *76*, 873, *883*
Shearin, S., 448, *457*
Shedroff, N., 134, *145*
Sheedy, E. J., 207, 218, *227*
Sheil, B. A., 7, *24*
Shelby, R., 446, *458*
Shelden, S., 735, *739*
Sheliga, B. M., 33, *40*
Shell, J., 181, 188, *198*
Shen, C., 171, *176*, 194, *198*
Shen, H., 288, *294*, 647, *656*, *658*, 1249, *1262*
Shepard, R. N., 45, 48, *61*
Shepherd, A., 575, *601*
Shepherd, M., 31, *40*
Sheppard, S., 1119, *1124*
Sheras, P. L., 770, *775*
Sheridan, C., 820, *828*
Sheridan, J. G., 270, *294*
Sheridan, K., 87, *91*
Sheridan, T. B., 574, 576, 581, 583, 589, *600*, *601*, 624, 630, *637*, *638*, *676*, 724, 727, 728, *738*, *739*, *740*
Sherman, L., 797, 798, 799, *806*, *807*
Sherman, R. M., 1245, *1261*
Sherwood, S., 285, *292*
Shewmaker, D. T., 1042, *1059*
Shi, B., 854, 859, 861, *869*
Shiffrin, R. M., 119, 120, 125, *132*
Shilling, R. D., 624, *638*
Shim, J. P., 668, *678*
Shimamoto, B., 623, *638*
Shimi, S. M., 37, *39*
Shin, D. J., 360, *378*
Shinn-Cunningham, B., 624, *638*
Shipley, T., 825, *827*
Shively, R. J., 724, *738*
Shmulevich, I., 491, *507*
Shneiderman, B., 7, *24*, 36, *40*, 163, 167, 169, *174*, *176*, 181, *198*, 274, *294*, 453, *457*, 497, *505*, 510, 515, 517*t*, 518, 530, *542*, *543*, 563, *572*, 824, *828*, 872, 874, 879, *882*, 1130, *1149*, 1152, 1155, *1169*, 1251, 1254, *1263*, *1265*
Shoemaker, G. B. D., 1253, *1265*
Shojania, K. G., 662, *677*
Sholapurwala, R., 874, *882*
Shore, D. I., 55, *60*
Short, J., 549, *557*
Shortliffe, E. H., 595, *601*, 668, *677*
Shouls, A., 1087, *1105*
Shoval-Katz, A., 403, *411*
Shriver, S., 422, 423, *431*
Shu, C., 642, *658*
Shu, K., 667, *678*
Shubina, A. M., 644, *658*
Shulman, L. S., 594, *598*
Shuster, J., 1123, *1124*
Shweder, R. A., 79, *91*, 1246, *1265*
Siang, S., 1259, *1262*
Sibert, L. E., 38, *40*, 168, *174*, 842, *852*
Sidel, P. H., 744, *759*
Sides, S. A., 624, *637*
Sidner, C. L., 417, *430*, 435*n*, 437, 438, *455*, *456*, *457*, 547, *558*
Siegal, D., 270, *294*
Siegel, D., 938, 939, *947*
Siegel, J., 548, 549, *555*, *556*, *557*
Siegman, J., 10, *23*
Siem, F. M., 631, *638*
Siewiorek, D. P., 296, 298, 305, 308, 311, *311*, *312*
Signorini, A., 561, *572*
Silberg, W. M., 670, *678*
Sillence, E., 652, 653, *658*
Silling, S. M., 764, *774*
Silsbee, P. L., 417, *432*
Silva, L., 865, *869*
Silver, D., 230, *244*
Silverman, B. G., 485, *487*, 593, *601*, 796, 805, *808*
Silverstein, B., 314, *325*
Sim, G., 797, *808*
Simis, A., 875, *883*
Simmel, D., 547, *556*
Simmonds, G. R., 723, *740*
Simmons, D., 877, *882*
Simms, S., 799, *807*
Simoff, S J., 1065, *1079*
Simon, H. A., 8, 18, *24*, 50, *60*, 66, *75*, *76*, 94, 96, *112*, 151, *157*, 498, 499, *507*, 580, 582, *599*, *600*, 798, 800, 801, *807*, 1139, 1140, *1147*
Simon, J. R., 36, *41*, 56, *61*
Simon, R. T., 99, *113*, 647, *658*
Simoneau, G., 168, *175*
Simonsen, J., 575, *597*, 982, *987*, 1062, 1065, *1078*, *1079*, 1250, *1264*
Simpson, A., 6, *23*
Simpson, B., 652, *655*
Simpson, H. M., 707, *718*
Simpson, K. T., 940, *947*
Simpson, N., 713, *719*
Sims, E., 889, *893*
Sims, V. K., 70, 71, *75*
Simske, S., 440, *458*
Sinclair, M., 162, 164*f*, 165, 171, *174*, 180, 181, *197*, 231, *245*
Singer, J. A., 1054, *1058*
Singer, J. E., 85, *91*
Singer, M., 631, *638*
Singh, G., 1212, *1219*
Singhal, S., 554, *557*, 626, *638*
Singletary, B. A., 590, *598*
Singleton, S., 643, *656*
Singley, M. K., 99, *112*

Sinha, R., 1015, *1016*
Siochi, A. C., 1086, *1105*, *1124*
Sipay, E., 876, *883*
Siroux, J., 414, 415, *432*
Sissors, J., 993, *1016*
Sit, R. A., 782, 787, *791*
Sjögren, D., 1065, 1066, 1069, 1071, 1072, 1073, *1078*
Sjomeling, M., 770, *775*
Skaburskis, A. W., 181, 188, *198*
Skattebo, A. L., 1055, *1058*
Skeels, C., 299, 310, *312*
Skelly, T., 87, *90*
Skelton, T. M., 1144, *1149*
Skepner, E., 168, *175*
Skilbeck, C. E., 833, *852*
Skinner, B. F., 802, 803, *808*, 1146, *1149*
Skinner, C. S., 138, *146*
Skinner, E. A., 83, *91*
Skirka, L., 576, 582, *601*
Skylar, A., 1133, *1149*
Skov, M. B., 794, 799, *806*, *808*
Slagter, R., 549, *558*
Slamecka, N. J., 53, *61*
Slater, A., 218, *227*
Slater, J., 1062, *1081*
Slater, M., 233, *245*, 631, *638*
Slaughter, S. A., 73, *75*
Slavin, R. E., 553, *557*
Sleeman, K. D., 823, *828*
Sleet, D., 703, *718*
Slivovsky, L. A., 178, *198*
Sloan, L. L., 859, *869*
Slobogin, P., 845, *852*
Slocum, J. W., 674, *678*
Sloman, A., 83, *91*
Sloss, D., 707, *718*
Slovic, P., 669, 673, *676*, *678*
Smailagic, A., 296, 305, 308, 311, *311*, *312*
Small, B. J., 814, *827*
Small, S., 876, *883*
Smart, L. J., 632, *638*
Smeaton, A., 565, *571*
Smelcer, J. B., 51, *61*
Smetters, D. K., 648, 650, *655*
Smiddy, W. E., 858, *869*
Smith, A., 180, *198*
Smith, B. A., 162, 168, *176*, 430, 442, 457
Smith, C. A. P., 44, *62*
Smith, D. C., 168, *175*, 470, 472, *486*, *487*, 804, *807*, 968, *987*
Smith, E. E., 1223, 1227, *1239*
Smith, G., 580, *600*
Smith, H. J., 643, *657*
Smith, H. T., 10, *24*
Smith, I. D., 773, *774*
Smith, I. E., 179, *197*, 414, 415, 426, 427, *429*, 548, *555*, 1033, *1038*, 1234, 1249, 1253, 1255, *1262*, *1263*
Smith, J., 583, *601*, 642, *658*, 780, *790*
Smith, J. E., 29, *40*
Smith, J. E. K., 50, *60*
Smith, J. R., 181, *198*
Smith, J. W., 578, *598*
Smith, K., 734, *740*
Smith, M. A., 171, 172, *174*, *176*, 492, *507*
Smith, M. E., 1223, 1234, *1238*, *1239*
Smith, M. J., 314, 315, 323, *324*, *325*, *326*, *598*
Smith, M. R., 1242, 1243, *1266*
Smith, N. W., 784, *792*
Smith, P. A, 1138, *1149*
Smith, P. J., 574, 576, 578, 579, 580, 583, 593, 594, 596, *598*, *601*
Smith, P. L., 48, *61*
Smith, R. B., 256, *264*, 276, *292*, 642, *655*
Smith, S. L., 4, 6, 11, 14, *24*
Smith, S. W., 644, *658*
Smith, T., 15, *22*
Smith, T. J., 57, 58, *61*
Smith, T. M., 1253, *1264*
Smith-Jackson, T. L., 360, *378*
Smither, J. A., 785, *790*
Smits, D., 570, *572*
Smutz, W., 168, *176*
Smyth, B., 442, 443, *456*, *457*
Snelling, L., 1065, *1081*
Snellings, P., 875, *883*
Snibbe, S. S., 168, *176*
Snoeck, B., 297, *312*
Snoeck, C., 297, *312*
Snow, C., 973, *987*
Snowling, M., 875, *883*
Snyder, C., 403, *411*, 1025, *1039*, 1135, *1149*
Snyder, M., 767, *774*
Sobral, J., 765, *774*
Soderston, C., 1139, *1149*
Soergel, D., 527, *543*
Sohn, C., 181, 188, *198*, 1255, 1263
Sokal, J. O., 830, *852*
Sokkappa, B. G., 735, *740*
Sokol, S., 496, *506*
Sokolov, J. L., 1052, *1059*, 1135, *1149*
Soller, A., 444, *457*
Soloway, E., 802, 804, *808*, 879, *882*
Solvberg, A. M., 765, *774*
Somberg, B. L., 55, *61*, 735, *739*
Sommer, A., 858, *867*
Sommer, D., 219, 220, *225*, *227*
Sommer, R., 171, *176*
Sommerich, C., 218, *227*
Sommerville, I., 583, *597*, 1046, 1055, *1059*
Somoza, M., 87, *91*
Sonderegger, P., 980, *987*
Song, A. W., 47, *59*
Songer, N. B., 71, *75*, 805, *808*
Sonnemanns, P., 481, *506*
Sonnenberg, F. A., 666, *678*
Soohoo, C., 143, *145*, 652, *656*
Soon, C., 365, *377*
Sorce, J. F., 1140, *1149*
Sorensen, S. J., 675, *677*
Soriyan, H. A., 617, *619*, 1250, *1264*
Sotoyama, M., 317, 318, 319, *325*
Souchon, N., 1119, *1124*
Soukoreff, R. W., 46, *60*, 168, *175*
Southard, D. L., 29, *40*
Spackman, M. P., 81, *91*
Sparck Jones, K., 566, *572*
Sparks, R., *1175*, 1189
Sparrell, C., 415, *430*
Spaulding, V. A., 782, *791*
Spehar, B., 491, *507*

Spence, C., 30, *41*, 420, 422, *429*
Spence, R., 519, 533, 537, *543*
Spencer, A., 576, 583, *601*
Spencer, B., 644, *655*
Spencer, H., 337, *353*
Spencer, R., 1173, 1176, *1189*
Spencer, S. J., 770, *774*
Spenke, M., 288, *292*
Spiekermann, S., 643, *658*
Spieler, D. H., 68, *76*
Spilka, G., 331, *353*
Spindle, B., 365, *378*
Spine, T. M., 1053, 1058
Spinellis, D., 283, *292*
Spinello, R. A., 1248, *1266*
Spink, A., 565, 566, *572*
Spitulnik, J., 804, *808*
Spool, J. M., 403, *411*, 903, *915*, 1135, *1148*
Sprafka, S. A., 594, *598*
Sprangers M. A. G., 669, *677*
Springer, J., 217, *227*
Sproull, L., 139, *146*, 547, 553, *555*, *556*, *557*, 1242, *1266*
Spurr, C. D., 667, *678*
Spyropoulou, E., 647, *658*
Squibb, H. R., 68, *76*
Squire, K., 608, *619*
Squires, S., 976, *988*
Srinivasan, M. A., 624, *635*
St. Amant, R., 108, 110, *112*, *113*, 575, *601*, 1201, *1207*
Stacey, D., 669, *677*
Stachel, B. B., 435*n*, 456
Stahl, S. A., 69, *76*
Stammer, R. B., 786, *792*
Stammerjohn, L. W., 314, 321, 322, *326*
Stammers, R. B., 150, *157*, 575, *597*, 1199, *1206*
Stamper, T., 171, *174*
Stanford, J., 143, *145*, 652, *656*
Stanney, K. M., 204, 223, *227*, 623, 624, 625, 626, 627, 628, 629, 630, 631, 633, *635*, *636*, *637*, *638*, 1222, 1223, 1223*n*, 1231, 1236, *1239*
Stanovich, K., 875, *883*
Stanton, D., 795, *808*
Stanton, N. A., 148, 150, 151, 152, 153, 154, 155, 156, *157*, *158*, 575, 597, 598, 1202, 1205, *1207*
Staplin, L., 704, *718*
Star, S. L., 1004, *1016*, 1070, *1081*, 1248, *1264*
Stark, L., *219*, 226
Starkes, J. L., 33, *39*
Starner, T., 168, *173*, 182, 185, *196*, *198*, 297, 300, 303, *311*, *312*, 451, *457*, 547, *557*, 890, *892*, *893*
Starr, B., 552, *554*
Starr, S. L., 736, *740*
Stash, N., 570, *572*
Stasko, J., 650, *656*
Staveland, L. E., 120, 127, *131*
Steed, A., 631, *638*
Steele, C. M., 766, 770, *773*, *774*, *775*
Steele, N., 944, *947*
Steers, R. M., 746, *759*
Steffen, J. L., 526, *542*
Steffensmeier, T., 764, *774*
Stefik, M., 276, *294*
Stefik, M. J., 540, *543*
Stein, B. E., 420, 422, *429*
Stein, R., 298, *312*
Steinberg, E. P., 858, *869*
Steinberg, L., 1216, 1217, *1219*
Steiner, D., 1067, 1068, *1078*
Steiner, T. J., 420, *430*
Steinfeld, A., 714, *718*, 888, *893*
Steinkeuhler, C., 608, *619*
Steinman, R., 873, *881*
Steinmetz, R., 1112, *1125*
Stelmach, G. E., 28, *41*
Stenz, A., 591, *602*
Stephanidis, C., 356, *380*, 823, *828*, 837, *852*, 859, 863, 864, *869*, *870*, 1085, 1088, 1098, 1100, 1101, 1102, *1105*, 1111, *1125*, 1254, *1266*
Stephens, E., 1144, *1149*
Stephenson, A., 1139, *1147*
Stepp, R., 567, *572*
Stern, K., 1135, *1149*
Sternberg, R. J., 813, *828*
Sternberg, S., 45, 46, 47, 48, 50, *61*
Stetson, J., 1138, *1149*
Steuer, J., 139, *146*
Steury, K., 748, 749, 750, 754, *758*, *759*, 1138, 1146, *1148*
Steve, M., 1212, *1219*
Stevens, A. L., 64, *75*, 576, *598*, 731, *738*
Stevens, W. R., 273, 281, *294*, 854, 865, *868*
Stevenson, M. 705, *718*, 877, 878, *882*, *883*
Stewart, A. J., 196, *198*
Stewart, C. D., 568, 569, *571*, *572*
Stewart, J., 795, 797, *807*, *809*
Stewart, P., 648, *655*
Stewart, T. F. M., 6, *22*, 207, *225*, 314, *324*
Stiel, H., 171, *175*
Stiens, S., 834, *852*
Stiensmeyer-Pelster, J., 771, *774*
Stifelman, L. J., 169, 172, *176*, 278, *294*, 475, *487*
Stiff, J. B., 141, *146*
Stille, A., 366, *378*
Stimson, M. J., 1139, *1148*
Stins, J. F., 57, *60*
Stinson, M., 891, *893*
Stivoric, J., 296, 298, *311*, *312*
Stockton, K., 863, *867*
Stoffer, T. H., 36, *41*
Stoffregen, T., 632, *638*
Stokes, M. J., 183, *196*, 624, *637*, 1197, *1208*
Stolz-Loike, M., 781, *792*
Stone, C. J., 183, 186, *196*, *197*
Stone, D. H., 1153, *1169*
Stone, J. D., 55, *60*, 770, *775*
Stone, M. C., 169, *173*, *174*, 1033, *1038*
Stone, P., 218, 219, *227*
Stone, R., 576, *601*, 633, *638*
Stonecipher, J. W., 726, *738*
Stork, D. G., 181, *198*, 414, 415, 420, 424, 426, *430*, *432*
Storms, R. L., 627, *638*
Storrøsten, M., 550, *557*
Strack, F., 84, *91*
Strass, P., 1033, *1039*
Strauch, B., 578, *602*
Straus, S. G., 548, *557*
Strayer, D. L., 706, *716*
Strecher, V. J., 138, *146*
Street, D. A., 858, *868*
Street, R. L., Jr., 138, *145*
Streeter, L. A., 73, *75*

Streitz, N., 1112, *1125*
Strickland, D., 625, *638*
Strieb, M. I., 1196, *1207*
Stroebe, W., 1022, *1038*
Strohm, P., 583, *598*, 594, *601*
Strom, B. L., 667, *677*
Strommen, E., 795, 796, 804, *809*
Stronge, A. J., 786, *791*
Strub, M. H., 1195, *1207*
Strybel, T., 249, *264*
Stuart, R., 12, *23*, 490, *506*
Stuckless, R., 891, *893*
Sturgeon, N. L., 1246, *1266*
Stutts, J. C., 704, *718*
Su, H., 533, *543*
Su, Q., 417, *432*
Su, V., 473, 475, *486*
Subbaram, M., 218, *227*
Subrahmanyarn, K., 772, *775*
Subramani, M., 439, *456*
Suchman, L., 581, *602*, 729, *740*, 984*n*, 966, 971, 972, 973, 980, 982, *986*, *987*, *988*, 1062, 1064, 1065, 1066, 1068, *1081*, 1250, 1255, *1266*
Sudman, S., 754, *758*, *759*
Suenaga, Y., 426, *429*
Sugihara, T., 232, *245*
Sugiura, A., 182, *198*
Suhm, B., 414, 419, *428*, *431*, *432*, 861, 863, *869*
Sujan, H., 81, *91*
Sujan, M., 81, *91*
Sukaviriya, P. N., 110, *111*, 171, *174*
Suleiman, M., 288, *294*
Suler, J., 604, *619*
Sullivan, C., 142, *146*
Sullivan, E., 474, *485*
Sullivan, J. W., 78, *89*, 169, *173*, 417, *429*, 633, *637*
Sulzby, E., 769, *774*
Sum, A., 874, *883*
Sumby, W. H., 415, *432*
Sumi, Y., 444, 445, *457*
Sumikawa, D., 248, *263*, 863, *867*
Summerfield, A. Q., 415, 420, *430*, *432*
Summerskill, S. J., 563, *571*
Sumner, E. E., 526, *542*
Sun, C., 288, *294*
Sun, D., 288, *294*
Sundali, J., 578, *597*
Sundblad, Y., 1065, *1078*
Sundeen, M., 704, *718*
Sunder, S., 548, *556*
Sung, H. Y., 663, *676*
Supattanasiri, G., 109, *111*
Suraweera, P., 445, *457*
Surendran, A. C., 435, *457*
Sutcliffe, A. G., 394, 398, 399, 401, 403, 406, 408, *410*, *411*, 781, *792*, 1046, 1055, 1056, *1058*, *1059*
Sutherland, I., 5, *24*
Suthers, D. D., 553, *556*
Sutton, R., 1195, *1207*
Suzuki, H., 474, *487*
Svanæs, D., 1068, *1081*
Svebak, S., 84, *92*
Svensson, M., 1254, *1262*
Swaminathan, K., 171, *176*
Swan, J. E. II, 630, *636*
Swan, L., 277, *294*
Swann, W. B., 767, *774*
Swanson, H. L., 876, *882*, *883*
Swartz, K. O., 630, *638*
Swartz, L., 550, *557*
Swearingen, K., 172, *176*
Swede, H., 785, *790*
Sweeney, L., 643, 654, *657*, *658*
Sweeney, T., 625, *635*
Sweitzer, G., 317, 318, 319, *325*
Sweller, J., 69, *74*, 397, *410*, 420, 422, *430*, *432*
Swenson, H. N., 736, *740*
Swenton-Wall, P., 966, *986*, 1066, *1077*
Swierenga, S. J., 1140, 1143, *1149*
Swinnen, S., 71, *76*
Swope, S., 1257, *1265*
Sycara, K., 583, *602*
Sykes, J., 683, *700*
Sykeso, P., 267, *293*
Szalma, J. L., 116, 117, 118, 120, 121, 122, 123, 126, 127, 128, 129, *131*, 132, *132*
Szczepkowski, M. A., 581, *601*
Szczur, M., 1119, *1124*
Szekely, P., 1123, *1124*
Szeliski, R., 162, *174*

T

Taatgen, N. A., 109, *113*
Tabert, J., 1253, *1262*
Tachi, S., 231, *244*, *245*
Tachibana, K., 182, *197*
Tahir, M., 941, *946*
Tai, B.-L., 54, *61*
Tait, A., 219, *227*
Tait, V., 669, *677*
Takada, H., 224, *226*
Takai, J., 224, *226*
Takayama, L., 1135, *1149*
Takeda, K., 565, *572*
Tal, R., 168, *174*, *175*
Talmi, D., 51, *61*
Tamassia, R., 531, *542*, *543*
Tammaro, S. G., 551, *556*
Tamminen, S., 299, *312*
Tan, A., 708, *718*
Tan, D. S., 406, *411*, 1253, *1266*
Tan, H. Z., 178, *198*
Tan, S. L., 630, *635*
Tanaka, K., 1138, *1149*
Tandy, S. B., 80, *89*
Tanenbaum, A. S., 547, *557*
Tang, A., 420, 422, *432*
Tang, J. C., 547, 548, *555*, *557*, 642, *655*, 1212, *1219*, 1244, 1249, 1253, 1255, *1263*, *1266*
Tang, K. P., 438*f*, 439, 456
Tanke, E. D., 767, *774*
Tapia, E. M., 182, *198*
Tarasewich, P., 318, *325*
Tarr, R. W., 633, *637*
Tashev, I., 181, *198*
Tasker, S., 262, *263*
Tatar, D., 276, *294*

Tauber, E. R., 143, *145*, 652, *656*
Tavani, H. T., 1248, *1266*
Taveira, A. D., 315, *324*
Taylor, A., 277, *294*
Taylor, A. S., 908, *915*
Taylor, F. W., 2, *24*
Taylor, J. G., 85, *89*
Taylor, J. M., 436, *456*
Taylor, L., 436, *458*
Taylor, R. G., 832, *852*
Taylor, R. M., 734, *739*, *740*, 1233, *1238*, *1239*
Taylor, R. P., 491, *507*, 803, *809*
Taylor, T. L., 33, *41*, 601, 607, *619*
Taymans, J., 877, *882*
Teasley, S. D., 551, 552, *557*, 1249, *1264*
Teater, P., 667, *677*
Tedesco, D., 1143, *1149*
Tees, R. C., 490, *507*
Teevan, J., 441*n*, *456*, *457*
Teich, J. M., 667, 668, *676*
Teichner, W. H., 49, *61*
Telang, R., 548, *556*
Telford, C. W., 35, *41*
Tellam, D., 1175, *1189*
Tellegen, A., 87, *92*, 128, *132*
Teltrow, M., 643, *658*
Temple, L., 764, 765, *775*
Tennent, P., 285, *292*
Tenner, E., 37, *41*
Tenneson, D., 192, *198*
Tepper, P., 415, *430*
Terrano, M., 751, 753*t*, *759*, 1146, *1148*
Terveen, L., 445, 448, *457*, 553, *555*
Terzopoulos, D., 415, *432*
Tesch-Romer, C., 64, *75*
Tessmer, M., 575, *599*
Tester, J., 134, *145*
Tetlock, P., 579, *602*
Tetroe, J., 669, *677*
Tetsutani, N., 843, *850*
Tettey, S., 218, *226*
Thackara, J., 1065, *1081*
Thacker, P. D., 634, *638*
Thagard, P. R., 56, *59*, 66, *75*
Thakor, N. V., 784, *791*
Thalmann, D., 625, *635*
Thalmann, N. M., 625, *635*
Tharp, V., 629, *638*
Thatcher, J., 863, *870*
Thayer, R. E., 84, *91*, 128, *132*
Theeuwes, J., 38, *39*
Theis, D., 209, 210*f*, 212, 214, 215, 216*f*, *227*
Theofanos, M., 1134, *1149*
Theoharakis, V., 6, *24*
Thevenin, D., 1108, 1119, *1124*, *1125*
Thiel, E. C., 669, *677*
Thiemann, S., 1166, *1169*
Thies, S., 450, *456*
Thimbleby, H. W., 248, *264*, 287, *294*
Thomas, A., 667, *677*
Thomas, B., 1135, *1149*
Thomas, C., 1075, *1078*
Thomas, D. W., 6, *24*, 365, *378*, 795, *807*
Thomas, F., 87, *92*
Thomas, J. C., 492, 496, 499, 500, 501, 503, 504, *506*, *507*, 1049, *1058*, 1251, 1254, *1266*
Thomas, J. R., 794, 795, *809*
Thomas, M., 578, *601*
Thomee, S., 318, *325*
Thomke, S., 976, *988*
Thompson, C. A., 138, *146*
Thompson, D. M., 624, *635*
Thompson, S. A., 399, *411*
Thomsen, D., 647, *658*
Thomsen, M., 475, *486*
Thomson, D. M., 53, *61*
Thomson, N., 52, *59*
Thoresen, M., 218, *225*
Thorisson, K., 86, *89*, 415, *430*
Thorndike, E. L., 632, *638*
Thornton, J. G., 668, 669, *676*
Thorpe, S. J., 725, *740*
Thorson, E., 81, *92*, 138, 142, *146*, 682, *700*
Thropp, J. E., 118, 120, 126, *131*, *132*
Thuraisingham, B., 643, *658*
Thurman, R. A., 625, *637*
Thurrott, C., 1135, *1149*
Tian, L., 642, *658*
Tian, Y.-L., 85, *92*
Tianfield, H., 593, 595, *602*
Tibshirani, R. J., 705, *718*
Tiedermann, J., 771, *775*
Tielsch, J. M., 858, *869*
Tierney, C., 707, *718*
Tierney, W. M., 667, *677*
Tiger, L., 684*t*, 685, *700*
Tijerina, L., 708, *718*
Tilley, A., 219, *227*
Tillotson, J., 277, *294*
Timminga, E., 138, *145*
Timpka, T., 1069, *1079*
Tinker, R., 803, *809*
Tipper, S. P., 29, 33, 34, 35, 38, *40*, *41*, 725, *740*
Tlauka, M., 50, *61*
Tobey, D., 575, *602*
Todd, P. A., 138, *146*, 669, *678*
Todman, J., 764, *775*, 818, *828*, 838, 840, *849*
Todorov, E., 634, *636*
Toffetti, A., 708, *717*
Tognazzini, B., 64, *76*
Togo, D. F., 674, *678*
Tohkura, Y., 424, *432*
Tollis, I. G., 531, *542*
Tolmie, P., 501, *506*
Tomaka, J., 121, *131*
Tomasi, C., 181, *198*
Tombari, M. L., 137, 138, *146*
Tombaugh, J. W., 162, *173*
Tombu, M., 56, *61*
Tomioka, K., 360, *378*
Tomita, F., 233, *244*
Tomkins, A., 564, *571*
Tomlinson, M. J., 417, 419, 420, *432*
Tooby, J., 79, *92*
Tørning, K., 846, *850*
Torobin, J., 610, *619*
Törpel, B., 1065, 1069, *1081*
Torreano, L., 496, *506*

Torunoglu, I., 181, *198*
Toth, J., 754, *758*
Toyama, K., 181, *199*
Toye, E., 270, *294*
Toyoshima, Y., 1054, *1058*
Tractinsky, N., 403, *411*
Tracy, W., 336, *353*
Tran, T., 1254, *1264*
Tran, Y., 841, *850*
Traugott, M. W., 1167, *1168*
Traverso, P., 581, *598*
Tredennick, N., 623, *638*
Trefftz, H., 626, *637*
Treisman, A. M., 30, *41*, 54, 55, *61*, 395, 409, *411*
Tremblay, L., 33, *41*
Trepess, D., 274, *292*
Trevor, J., 171, *176*, *292*
Trewin, S., 436, 437, 450, *457*, 839, 845, 848, *852*
Trigg, R. H., 274, *293*, 971, 973, 980, 982, *986*, *988*, 1065, 1066, 1068, 1075, *1081*
Trimble, J., 171, *176*
Trimmel, M., 47, *61*
Trompenaars, F., 365, *380*
Truran, M., 567, *572*
Tryggvason, J., 889, *892*
Tsai, W., 972, *987*
Tsandilas, T., 441, 446, 450, *458*
Tsapatsoulis, N., 85, 86, *89*
Tscheligi, M., 1065, *1081*
Tschudy, M. W., 1069, 1070, 1073, *1081*
Tseng, H., 134, 141, *145*, 1254, *1262*
Tseng, S., 134, 141, *146*
Tsimhoni, O., 563, *572*, 709*t*, *717*
Tucker, M., 36, *41*
Tudor, L. G., 1045, *1059*, 1069, 1070, *1080*, *1081*
Tufte, E. R., 403, *411*, 517, 517*t*, 518*f*, 537, *543*, 576, 580, *602*
Tukey, J. W., 517, *543*
Tullis, T. S., 55, *61*, 101, *113*, 1137, 1138, 1143, *1149*
Tulving, E., 53, *61*
Tumanon, R. C., 830, *852*
Turban, E., 574, *602*
Turiel, E., 1246, 1252, 1254, *1264*, *1266*
Turieo, M., 624, *637*
Turing, A., 8, *24*
Turisco, F., 675, *678*
Turk, M., 414, 417, *432*, 625, *638*, 859, 861, *870*
Turkki, L., 800, *807*
Turkle, S., 1256, *1266*
Turner, C. H., 365, *380*
Turner, C. S., 1252, *1264*
Turner, T. J., 79, *91*
Turow, J., 652, *658*
Tutt, M., 889, *892*
Tversky, A., 57, *60*, 578, 580, 595, *598*, *599*, *602*, 1043, *1058*, *1059*
Tweedie, L. A., 533, 536, *543*
Tweney, R., 579, *600*
Tygar, J. D., 647, 650, 651, *656*, *658*
Tyrell, R., 218, *225*

U

Ucros, C. G., 81, *92*
Ueno, S., 439, 440*f*, *456*
Ulfarsson, G., 1255, *1266*
Ullman, E., 703, *717*
Ullmer, B., 170, *174*, 194, *198*, 284, *294*, 470, 473, 474, 475, 477, 478, 484, *486*, *487*, 795, *808*, 1036, *1038*
Ulrich, R. S., 1255, *1266*
Umiltà, C., 33, 36, *40*, *41*
Ummelen, N., 1140, *1148*
Underkoffler, J., 470, 475, 478, 484, *487*
Underwood, B. J., 52, *60*
Underwood, G., 702, *716*
Underwood, P., 500, *507*
Underwood, S. E., 706, *719*
Upshaw, L., 993, *1016*
Upton, E., 270, *294*
Uren, E., 938, *947*
Urey, H., 204, *227*
Uribe, T. E., 436, *456*
Urokohara, H., 1138, *1149*
Usher, J., 19, *23*
Usher, M., 49, *61*
Utsui, Y., 714, *718*
Uyeda, K. M., 1178, *1188*

V

Vaghi, I., 278, *293*
Väinämö, S., 1070, *1079*
Vaknin, S., 607, *619*
Valacich, J. S., 550, *557*
Valle-Inclán, F., 50, *61*
Valleau, M., 417, *429*, 840, *850*
Vallone, R., 578, *598*
van Baar, J., 171, *175*
Van Breda, L., 1195, *1207*
Van De Moortele, P. F., 1234, *1238*
Van den Bergh, J., 1123, *1124*
van den Besselaar, P., 1063, 1077, *1081*
van der Heiden, G. H., 321, *325*
van der Mast, C. A. P. G., 634, *636*
van der Veer, G., 549, *558*
Van Der Wege, M., 108, *111*
van der Zaag, C., 634, *637*
van Deusen, M., 1047, *1057*
van Dijk, T. A., 67, *76*
Van Doorn, J., 581, *600*
van Duin, R., 1022, *1038*
Van Durm, R., 864, *869*
van Gelderen, A., 801, *883*
van Gelderen, T., 875, *808*
van Gent, R., 419, *431*
Van Gorp, P., 1117, *1125*
Van House, N. A., 1249, *1266*
van Kesteren, I. E., 801, *809*
van Kleek, M., 547, *557*
van Leeuwen, L., 796, *808*
van Melle, W., 165, *175*, *551*, *556*
Van Mulken, S., 399, *410*
van Reekum, C., 116, *132*
van Schaik, P., 219, *227*
Van Zandt, T., 29, *40*, 44, 48, 57, *60*, *61*
Vander Zander, B., 1035, *1039*
Vanderdonckt, J., 1119, 1123, *1124*, *1125*
Vanderhagen, G., 701, 705, *717*

Vanderheiden, G. C., 843, 847, 848, *851*, *852*, 864, *870*, 887, *893*, 1140, *1148*
Vanderheiden, P. B., 818, *828*
Vanderheyden, P. B., 837, 845, *851*
Vanderplas, J. M., 724, *739*
Vandervelpen, C., 1123, *1124*
Vandevenne, H. F., 735, *740*
Vandor, B., 593, *600*
VanLehn, K., 446, *458*
VanRullen, R., 725, *740*
Varian, H. R., 267, *293*, 553, *557*
Varma, C., 142, *145*, 279, *292*, 652, *656*
Vasara, K., 1054, *1059*
Vatikiotis-Bateson, E., 415, 420, *431*, *432*
Vatrapu, R., 1142, 1143, *1149*
Veinott, E., 549, *557*, 1254, *1266*
Veitengruber, J. E., 732, *740*
Velger, M., 310, *312*
Velichkovsky, B. M., 53, *59*
Vellutino, F., 876, *883*
Venkataraman, S. T., 667, *676*
Venolia, G. D., 1253, *1263*
Ventura, D., 634, *635*
Venturi, G., 1172, *1189*
Vera, A. H., 1198, *1207*
Verbyla, J. L. M., 564, 565, *571*, *572*
Vergo, J., 318, *325*, 414, *431*
Verhaeghen, P., 54, *61*
Vermeeren, A., 1181, *1189*
Vermeeren, A. P. O. S., 801, *809*
Vernier, F. D., 194, *198*
Vernon, P. A., 794, *808*
Verplank, B., 137, *146*, 1135, *1148*
Verplank, W. L., 17, 22, 472, *486*, 589, *601*, 1049, *1059*
Vertegaal, R., 181, *198*, 549, *557*, *558*
Ververs, M. P., 730, *740*
Ververs, P. M., 732, *740*, 1228, *1238*
Vessey, I., 15, *24*
Vicente, K. J., 123, *130*, *132*, 171, *174*, 575, 576, 577, 581, *597*, *598*, *602*, 723, *740*, 787, *792*
Vickers, B., 359, *378*
Victor, D. A., 365, *378*
Vidal, F., 51, *59*
Viding, E., 54, *60*
Vidot, N., 288, *294*
Vidulich, M. A., 395, *411*, 420, *432*, 734, *740*
Viégas, F. B., 552, *558*, 606, *619*
Vigliani, A., 6, *23*
Viirre, E. S., 630, *636*
Vijaya Satya, R., 624, *636*
Villanuela, M. B. G., 317, *325*
Villar, N., 449, *455*
Vilnai-Yavetz, I., 903, *915*
Vince, J., 623, *638*
Vincent, D., 736, *740*
Violanti, J. M., 705, *719*
Virtanen, K., 1070, *1079*
Virzi, R. A., 1052, *1059*, 1135, 1140, 1142, *1149*
Vitali, F., 560, *571*
Vitense, H. S., 854, 854*f*, *868*, *870*
Vivacqua, A., 445, *458*
Viviani, P., 166, *176*
Vlissides, J., 1033, 1034, *1038*, *1039*
Vo, M. T., 415, 426, 427, *432*
Voelker, D., 79, *92*
Vogel, D. R., 15, *23*, *24*, 170, 171, *176*, 182, 191, *198*, 550, *557*
Vogler, C., 890, *893*
Voida, A., 1255, *1266*
Vollrath, D. A., 1077, *1078*
von Ahn, L., 692, *699*
Von Korff, M., 663, *678*
Von Thaden, T., 579, *602*
von Waldkirch, M., 204, 204*f*, 205*f*, 216, *227*
Vora, P., 1136, *1149*
Voss, M., 218, *225*
Votsis, G., 85, 86, *89*
Vranish, J. M., 181, *198*
Vredenburg, R., 908, *915*
Vries, G. D., 795, *808*
Vu, K. L., 28, *40*
Vu, K.-P. L., 38, *41*, 49, 50, 54, *60*, *61*
Vygotsky, L., 614, *619*
Vyzas, E., 85, *91*

W

Wachtel, P. L., 129, *132*
Waddell, P., 1255, *1266*
Waddington, P., 1087, *1105*
Wade, D. T., 833, *852*
Wadge, W. W., 568, *572*
Wagenaar, W. A., 148, *158*
Wager, K. A., *666*, 667, *678*
Wagner, C., 233, *245*
Wagner, D., 646, 657, *658*
Wagner, E. H., 663, *678*
Wagner, I., 1063, *1078*, *1079*
Wahlin, Å., 814, *827*
Wahlster, W., 414, 415, *432*, 439, *456*
Waibel, A., 415, 426, *429*, *432*
Wainer, H., 450, *458*
Wainryb, C., 1246, *1266*
Wakeman, I., 270, *293*
Wald, E., 846, *849*
Wald, M., 891, *892*
Waldrop, M. M., 4, *24*
Walendowski, A., 549, *556*
Wales, R., 171, *176*
Walker, B. N., 261, *264*
Walker, D. F., 804, *809*
Walker, M., 1135, *1149*
Walker, N., 51, *61*, 782, 784, *791*, *792*, 814, *828*
Walkerdine, J., 283, *293*
Wall, P., 971, *986*
Wallace, P., 604, *619*
Wallace, R. J., 29, 36, *41*
Wallace, V., 165, *173*
Wallace, V. L., 7, *23*
Wallbruch, R., 864, *867*
Waller, A., 817, 819, *827*, *828*
Wallin, J. S., 764, *775*
Walston, C. E., 502, *507*
Walters, J., 84, *92*
Walters, T. B., 541, *543*
Waltz, D., 580, *602*
Wang, C., 890, *893*
Wang, H., 49, *61*

Wang, H. C., 606, *619*
Wang, J., 426, *432*
Wang, M., 427, *429*
Wang, M. Q., 426, *431*
Wang, R. W., 593, *602*
Wang, X., 475, *486*
Wang, Y., 171, *175*, 475, *486*, *487*, 593, 595, *599*
Want, R., 170, 171, 172, *174*, *176*, 180, 182, *198*, 262, *264*, 643, *658*
Ward, A., 180, *198*
Ward, D. M., 666, *678*
Ward, K. G., 52, *59*
Ware, C., 180, 184, *197*, *198*, 310, *312*, 519, *543*
Warkentin, M., 668, *678*
Warm, J., 582, *598*
Warm, J. S., 45, *61*, 116, 117, 118, 119, 120, 122, 123, 125, 126, 127, 128, 129, *131*, *132*
Warren, S. A., 642, *658*
Warshaw, P. R., 15, *22*, 673, *676*
Washburn, D., 624, *636*
Wasinger, R., 451, *456*
Wason, P., 579, *602*
Wasserman, A. I., 1042, *1059*
Wasson, C., 966, 971, 973, *988*
Watanabe, T., 224, *226*
Waters, J., 649, *658*
Waters, K., 139, *146*, 553, *556*
Waters, S., 1144, *1149*
Waterworth, J., 861, *867*
Waterworth, J. A., 861, *870*
Watkins, M. J., 53, *59*
Watkins-Malek, M. A., 764, *774*
Watson, B. A., 634, *638*
Watson, D., 87, *92*, 128, *132*, 1233, *1239*
Watson, F. L., 33, *41*
Watson, R. S., 667, *676*
Wattenberg, M., 530, *543*, 552, *558*, 606, *619*
Watts, L., 549, *555*
Watzman, S., 576, *602*
Waugh, N. C., 492, *506*, *507*
Wayman, J., 182, *198*
Wayne, J., 873, *881*
Weaver, B., 33, *41*
Weaver, J. L., 118, 126, 127, 129, *131*
Weaver, K. D., 764, 765, *774*
Weber, G., 260, *264*, 863, *867*, *869*
Weck, M., 217, *227*
Weckert, J., 1257, *1266*
Weeks, D. J., 28, 33, 34, 35, 36, *39*, *40*, *41*
Weerasinghe, A., 445, *457*
Wegerbauer, C., 737, *740*
Weghorst, S., 139, *145*, 625, *638*
Wegner, D. M., 74, *76*, 81, 83, *92*
Wei, J., 1166, *1169*
Weibelzahl, S., 451*n*, *456*
Weidenhaupt, K., 1046, 1057, *1059*
Weidenman, N., 217, *226*
Weiffenbach, J. W., 784, 791
Weil, M. M., 633, *636*, 764, *775*
Weiler, E. M., 120, *132*
Weimer, D., 1254, *1263*
Weinberg, G. M., 7, *24*, 498, *507*
Weiner, B., 771, *775*
Weiner, J., 873, 876, *883*
Weinshenk, S., 384, *391*, 1276, *1277*
Weinstein, A., 993, *1016*
Weir, P. L., 35, *41*
Weiser, M., 171, *176*, 470, 485, *487*, 1036, *1039*, 1256, *1266*
Weiss, S., 224, *227*
Weissman, C., 170, *173*
Weitzman, L., 394, *410*
Weizenbaum, J., 1242, 1259, *1266*
Welch, B., 550, *555*
Welch, G., 183, *198*
Welch, R. B., 629, *638*
Welch, V., 650, *655*
Weld, D. S., 450, *458*
Welford, A. T., 28, 29, *41*
Well, A., 723, *740*
Weller, S. C., 972, *987*
Wellman, B., 610, 611, *618*, *619*, 643, *657*
Wellner, P., 181, *198*, 475, 478, *487*, 1036, *1039*
Wells, G., 578, 595, *597*
Wells, M., 889, *892*
Wells, R., 313, 323, *325*
Welsh, D., 1146, *1148*, *1172*, 1189
Welsh, T. N., 33, 34, 35, *38*, *41*
Welty, B., 653, *658*
Wempe, T. E., 723, *739*
Wen, J., 567, *572*
Wen, Z., 444*n*, 458
Wendell, J. B., 951, *963*
Wendt, K., 547, *555*
Wendt, P., 780, *790*
Werker, J. F., 490, *507*
Wertheimer, M., 1043, *1059*
Wesson, R., 422, *431*, *432*
West, D., 875, *883*
West, R., 875, *883*
West, S., 858, *869*
Westerlund, B., 807
Westerman, S. J., 786, *792*
Wetzel, I., 1066, *1079*
Wexelblat, A., 446, 447, *458*
Wexner, L., 84, *92*
Weymouth, T., 557
Whalley, S. J., 152, *158*
Wharton, C., 95, *112*, 575, 576, *599*, 1047, *1059*, *1080*, 1175, 1176, 1178, *1188*, *1189*, *1190*
Wheeler, S., 169, *176*
Wherry, E., 171, *176*
Wherry, R. J., 1196, *1207*
Whitcher, A. L., 973, *987*
White, A. W., 666, *678*
White, E. A., 1046, *1059*, 1065, 1066, *1080*
White, M., 492 *505*
White, S., 1256, *1265*
Whitecross, S., 1230, *1239*
Whitehurst, H. O., 723, *740*
Whiteman, M. C., 121, *131*
Whiteside, J. A., 934, *947*, 1026, *1038*, 1053, *1058*, 1199, 1200, *1208*
Whitfield, D., 733, *740*
Whiting, B., 970, 971, *988*
Whiting, J., 970, 971, *988*
Whitley, B. E., Jr., 765, *775*
Whitlow, S. D., 1228, *1238*
Whittaker, S., 547, 548, 549, *556*, *557*, *558*
Whittaker, S. G., 865, *870*
Whitten, A., 647, 650, *658*

Whittington, J., 99, *113*
Whitty, A., 1251, *1263*
Wholey, D. R., 74, *75*
Whyte, W. F., 971, 973, *988*
Wibom, R., 324
Wichansky, A., 1135, *1149*
Wichansky, A. M., 1139, *1148*
Wickelgren, W. A., 46, *62*
Wickens, C. D., 55, 58, *62*, 120, 121, 123, 124, 126, 127, 129, *132*, 148, 149, 150, *158*, 168, *176*, 219, *227*, 299, 310, *312*, 395, *411*, 420, 424, *432*, 449, *458*, 574, 575, 576, *600*, *602*, 632, *638*, 705, *717*, 724, 725, 728, 730, 732, 735, *739*, *740*, 1227, *1239*
Wickens, D. D., 52, 62
Widdel, H., 218, *227*
Wiedenbeck, S., 649, 652, *656*, *658*
Wiedenmaier, S., 219, 224, *226*
Wiederhold, B. K., 138, *145*, *146*
Wiederhold, M. D., 139, 145, *146*
Wiegmann, D., 578, 579, *602*
Wiener, E. L., 582, 590, *602*, 728, 729, *740*
Wiener, L., 1055, *1060*
Wiener, N., 1242, 1259, *1266*
Wierwille, W. W., 705, 715, 715*t*, *717*, *719*
Wierzbicka, A., 79, *92*
Wiesmeyer, M., 101, *113*
Wigdor, D., 168, *176*
Wiklund, M. E., 1135, *1149*
Wilbur, S., 548, 549, *555*, *557*
Wilcox, S., 888, *892*
Wilde, G. J., 439*n*, 458
Wildemuth, B., 879, *882*
Wilder, G., 764, *775*
Wildman, D. M., 1045, *1059*, 1065, 1066, *1080*
Wilensky, R.,
Wilensky, R. L., 437*n*, *456*, 581, *602*
Wilhelm, A., 270, *294*
Wilkens, M., 1254, *1263*
Wilkerson, B., 1055, *1060*
Wilkerson, M., 650, *657*
Wilkins, A., 219, *227*
Wilkins, R., 565, *571*
Wilks, A. R., 531, *542*
Wilks, Y., 444, *455*
Willeges, R. C., 726, *737*
William, C., 1067, 1068, *1078*
Williams, B., 1246, *1266*
Williams, D., 788, *790*
Williams, E., 548, 549, *557*, *558*
Williams, E. S., 674, *677*
Williams, J. C., 152, *158*
Williams, J. H., 742, *759*
Williams, K., 575, *602*
Williams, S., 1123, *1124*
Williamson, C., 497, *505*
Williges, R. C., 787, *792*, 1141, *1147*, 1177, 1178, 1185, 1186, *1188*
Willis, G. J., 531, *542*
Willis, S. L., 780, 785, *791*, *792*
Willis, T., 845, *852*
Willkomm, B., 632, *636*
Willows, D. M., 822, *828*
Willwacher, T., 171, *175*
Wilpon, J. G., 383, *391*
Wilson, A. D., 165, 167, 170, 171, *173*, *176*, 181, 188, 190, *198*, 816, *828*
Wilson, C. E., 1139, *1149*
Wilson, D. H., 182, *198*
Wilson, F. R., 172, *176*
Wilson, G. F., 625, *638*, 1222, 1223, 1225*t*, *1238*, *1239*
Wilson, J. M., 74, *75*, 154, 155, *158*
Wilson, K. R., 764, *775*
Wilson, P., 6, *23*
Wilson, S., 614, *618*
Wine, D. W., 204, *227*
Winkelholz, C., 218, *227*
Winner, L., 1062, *1081*, 1243, *1266*
Winograd, T., 171, *174*, 548, *558*, 804, *809*, *869*, *1125*, 1244, 1250, 1255, *1261*, *1264*, *1266*
Wirfs-Brock, R., 1045, 1055, *1060*
Wise, J. A., 491, *507*
Wisneski, C., 194, *197*, 475, *486*
Wisneski, G., 284, *294*
Wisniewski, L., 863, *869*
Withgott, M., 474, *486*
Witkin, B. R., 575, *602*
Witmer, B., 631, *638*
Witten, I. H., 287, *294*
Wittgenstein, L., 730, *740*
Wittig, F., 440, *457*
Wixon, D. R., 751, 753*t*, *759*, 909, *916*, 928, 930, 934, 936, *947*, *1038*, 1063, *1081*, 1146, *1148*, 1172, 1184, *1189*, *1190*, 1199, 1200, *1208*
Wobbrock, J., 840, 845, *851*, *852*
Wobbrock, J. O., 318, *325*
Woda, A. B., 765, *774*
Wogalter, M. S., 593, *602*
Wohldmann, E. L., 53, *59*
Wolf, C., 172, *176*, 384, *391*, 568, *572*
Wolf, C. G., 1130, *1149*
Wolf, E., 219, *227*
Wolf, G., 1062, *1078*, 1250, *1262*
Wolf, M., 968, *988*
Wolf- Kelly, K., 218, *226*
Wolfe, J. M., 55, *62*
Wolfe, R. J., 563, *572*
Wolfman, S. A., 450, *458*
Wolgast, E., 218, *225*
Wong, B., 297, *312*
Wood, A., 481, *486*, 794, *807*
Wood, C., 426, 427, *432*
Wood, D., 385, *391*, 1033, *1038*
Wood, L. E., 54, *62*
Wood, S., 951, *963*
Wood, S. D., 104, 110, *111*, *112*, 1199, 1205, *1207*, *1208*
Wood, T. O., 858, *866*
Woodcock, A., 1063, *1081*
Woods, D. D., 574, 576, 582, *599*, *601*, 734, 735, *739*, *740*
Woodson, W. E., 202, *227*
Woodward, J. P., 139, *146*
Woodward, M., 705, *718*
Woodworth, R. S., 632, *638*
Woolgar, S., 968, *987*
Woolrych, A., 1131, *1147*, 1172, 1173, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, *1188*, *1190*
Word, C. O., 769, *775*
Worden, A., 784, *792*, 814, *828*
Worrall, L., 84, *90*
Wren, C., 180, *198*
Wright, C. E., 29, *40*, 50, *60*, *61*

Wright, J. H., 31, *39*
Wright, L., 668, *677*
Wright, M. H., 547, *555*
Wright, P., 684*t*, *700*, 729, *738*, *740*
Wright, P. C., 17, *22*, 194, *196*, 257, *263*, 854, *867*
Wright, R. D., 33, *41*, 181, *196*
Wright, Z., 1254, *1261*
Wu, H., 864, *867*
Wu, L., 415, 427, *431*, *432*, 863, *869*
Wu, M., 171, *176*, 194, *199*, 651, *657*, 1077, *1081*
Wu, Y., 181, 188, *199*
Wullert J. R. II, 207, 207*f*, 209, 214, 215, *226*
Würz, W., 863, *867*
Wyatt, J. C., 670, 671, *678*
Wyer, R. S., 81, *89*

X

Xia, S., 288, *294*
Xiao, B., 422, 428, *431*, *432*
Xiao, L., 1065, *1080*
Xiao, Y., 310, *312*
Xue, P., 415, *429*

Y

Yacoob, Y., 85, *92*
Yacoub, S., 440, *458*
Yale, J. F., 674, *677*
Yamada, H., 168, *176*
Yamagata-Lynch, L. C., 617, *618*
Yamao, T., 213, *245*
Yan, J., 649, *658*
Yanagisako, S., 968, *988*
Yang, B., 171, *175*
Yang, C. S., 566, *572*
Yang, H., 554, *558*
Yang, J. Y., 415, *429*, *432*, 606, *619*
Yang, S., 674, *677*, 792, 840, *851*, *882*
Yankelovich, N., 387, 389, *391*, *557*, 642, *655*, 862, *870*
Yantis, S., 55, *62*, 576, *602*
Yardley, J., 357, 365, *379*
Yarin, P., 475, *486*
Ye, Y., 406, *410*
Yedidia, J. S., 474, *485*
Yee, K., 648, *658*
Yee, K.-P., 171, 172, *176*
Yee, N., 607, *619*, 772, 773, *775*
Yeh, Y., 120, 127, *132*
Yeh, Y.-Y., 219, *227*
Yelland, N., 764, *775*
Yenisey, M., 648, *658*
Yenni, K. R., 725, *737*
Yerkes, R., 121, *132*
Yeung, B., 840, *851*
Yi, J., *311*, *867*
Yli-Harja, O., 491, *507*
York, J., 442, *457*
York, W., 541, *542*
Yoshida,Y., 240, *244*
Yoshitake, R., 317, 318, 319, *325*
Young, B. C., 442, *455*
Young, D. E., 71, *76*
Young, M. A., 830, *852*
Young, M. J., 81, *90*
Young, M. S., 155, 156, *158*
Young, P. M., 44, *62*, 625, *638*, 1222, 1231*n*, 1223, 1236, *1239*
Young, R. M., 58, *59*, 65 *76*, 94, 99, 101, 105, 110, *111*, *112*, *113*, 713, *719*, 1049, 1057, *1058*, *1060*, 1130, *1149*,
Young, S., 769, *774*
Youngme, M., 139, *146*
Youngs, E., 1249, *1265*
Yu, C. T., 566, *572*
Yu, K., 204, *225*
Yu, W., 862, *870*
Yua, K., 204, *225*
Yuen, M., 744, *759*
Yuille, J. C., 492, *506*

Z

Zabowski, D., 436, *457*
Zacharia, G., 611, 612, *619*
Zachary, W., 1197, *1208*
Zacks, R. T., 68, *76*
Zajonc, R. B., 84, *91*
Zakaria, M. R., 568, 569, 570, *571*, *572*
Zaklad, A. L., 1196, *1207*
Zancanaro, M., 454, *456*
Zander, A., 583, *597*
Zandri, E., 786, *792*
Zanna, M. P., 769, 770, *775*
Zaphiris, P., 611, 612, 614, *618*, *619*
Zarcadoolas, C., 878, *883*
Zasler, N. D., 832, *850*
Zedaker, M. S., 146
Zeleznik, R., 192, *197*
Zeller, C., 222, *226*
Zhai, S., 162, 163, 166, 166*f*, 167, 168, 170, *172*, *173*, *174*, *176*, 181, 186, 188, *199*, 414, 417, *430*, *432*, 843, 849, *852*
Zhang, D., 428, *430*
Zhang, H., 567, *572*, 579, *602*
Zhang, J., 582, *602*, 652, *658*, 726, *740*
Zhang, P., 6, 16, *24*, 901, *916*
Zhang, S., 169, *175*
Zhang, Y., 1154, *1169*
Zhang, Z., 180, *199*
Zhao, D., 890, *892*
Zhao, S., 169, *176*
Zhao, W., 185, *199*
Zheng, J., 553, *558*, 1254, *1266*
Zheng, X. S., 439, *456*
Zhou, M. X., 443, 444*n*, *458*
Zhu, E., 610, *619*
Zhuang, J., 230, *244*
Zickus, W. M., 837, 845, *851*
Ziefle, M., 208, 218, 219, 220, 220*f*, 221, 221*f*, 222, 224, *225*, *226*, *227*
Ziegler, C., 448, *458*
Zietsman, A. I., 139, *146*
Zillmann, D., 84, *92*
Zimand, E., 634, *636*
Zimmerman, B. J., 122, *132*
Zimmerman, T. G., 267, *294*
Ziomek, J., 717
Zirk, D. A., 582, *599*

Zittel, R. C., 634, 635, *638*
Zizi, M., 1112, *1125*
Zloof, M., 503, *507*
Zlot, R., 591, *602*
Zorich, B. S., 859, 861, *868*
Zou, Y., 204, *225*
Zubek, R., 684*t*, 700
Zuber, M., 1172, *1189*
Zuboff, S., 581, *602*
Zuckerman, O., 474, *487*
Zukerman, I., 439, *458*
Zurko, M. E., 647, *658*
Zwarun, L., 142, *146*
Zyda, M., 204, 223, *227*, 554, *557*, 623, 624, 625, 626, *638*

SUBJECT INDEX

A

Abstractions, 536–539
  data-based, 536–537
  methods, 536–539
  view-based, 537–539
Affect, 80, 182–183
  assessment, 81–82
  attention, 80–81
  effects of, 80–82
  measurement, 84–88
  memory, 81
  performance, 81
Aging and computer task performance, 786–787
Ambient intelligence, 1096–1103
  abstract interaction, object library, and polymorphic alternatives, 1102–1103
  ambient user interfaces, unified, 1097–1100
  instructions, ambient, 1097
  remote wireless GUI service, 1100–1101
  text case applications, 1101–1102
Architectures, cognitive, 94–95
  definitions, 94
  relevance to human-computer interaction, 95–96
  Unified Theories of Cognition, 94
Assessing and scoping methods, 1177–1182
  thoroughness, validity, and effectiveness, 1177–1178
  UIM assessment and comparison, improving, 1180–1182
  UIM assessment, methodological issues in, 1179–1180
  UIMs, other quantitative measures for, 1178
  UIMs, qualitative analysis of, 1179
Attention, 299–300
  Augmented Cognition project, 300
  mobile workers, monitoring of, 300
  office tasks, relation to users' environment, 299
  reduction of human effectiveness by multiple activities, 299
Auditory output, applications of, 259–263
  GUIs, sonic enhancement of, 259
  mobile and ubiguitous computing, sound for, 261–263
  sonification, 261
  users with visual impairments, sound for, 260–261

B

Benefits, quantifying, 1267
Browsing and linking, 563–565

C

Choice reaction tasks, information-processing in, 48–51
  response, 48–51
    execution, 50–51
    selection, 48–50
  stimulus identification, 48
Complexity and related concepts, 493–498
  complexity of system vs. contextual complexity, 497
  complexity of system vs. task complexity, 496–497
  complexity, feedback and interactivity, 497–498
  complexity, relationships of, 493–496
    measurement, 502
    naming scheme, 495–496
    nature of elements, 495
    nonlinearity, 495
    number, 494–495
    obscurity, 496
    simplicity, 493–494
    uncertainty, 494
    use, ease of, 493
Computer systems for older students, designing, 787–789
  hardware and software considerations, 787–789
Computer workstations, design of, 313
  auditory environment, the, 322–323
  chair as a critical element of the workstation, 319–320
  ergonomic implementation issues, 323–324
  ergonomic perspective, 314–315
    users' complaints, 314
  fixed computer workstation design features, guidelines for 316
    design studies, 315
  pids, carry along, and wearable IT, 318–319
    working conditions that lead to awkward posture, 315
  visual environment, 321–322
  workstation considerations, 321

Computer workstations, design of, (*continued*)
workstation for use with laptop computers, design, 317–318
chair and desk, 317
keyboard, 317
nonkeyboard input devices, 317
recommendations to reduce undesirable effects of poor workstation, 318
work environment and workstation layout, 317
working posture, 317
Computers and credibility, 140–143
credibility, importance of, 141
Web credibility research and guidelines for design, 142–143
computers as knowledge repositories, 141
computer use in measurements, 141
reporting on work performed, 141
reporting of own state, 141
running of simulations, 141
virtual environments, rendering of, 142
Computing systems designed to persuade, ethics of, 144
Constructions, 1073–1076
artifacts, descriptive, 1074
language, 1073–1074
low-tech prototypes, 1074–1075
prototyping, 1075
evolutionary, 1075–1076
cooperative, 1075–1076
Contemporary architectures, 102–110
ACT-R 5.0, 106–109
theory of information foraging, 108
comparisons, 109
EPIC, 103–106
LICAI/CoLIDeS, 102–103
Conversion rates, increased, 1273–1274
Cooperative work, groupware and computer-supported, 545
Cost justification, 1266
return on investment, 1266
Cost-justification approach to Web interfaces, importance of, 1269–1270
Cost-justification tool, spreadsheet model for, 1270–1273
financial metrics, 1272
usability cost justification, 1270
usability-engineering support, cost of, 1271
usability engineering support, cost of, 1271
Costs, quantifying, 1267
Culture dimensions, 364–376
culture as an issue for UI designers, 365–366
design issues, 374–376
future research issues, 376–377
Hofstede's dimensions of culture, 366–368
individualism vs. collectivism, 368–369
long- vs. short-term time orientation, 373–374
masculinity vs. femininity, 369–371
uncertainty avoidance, 371–373
Customer savings, 1273

D

Data-driven personas in product planning, design, and development, use of, 991
persona lifecycle method, creation of, 994
personal lifecycle, 994–1014
adulthood, 1006–1012
birth and maturation, 1003–1006
care team, creation, 995–996
conception and gestation, 997–1003
data sources and collecting data, 996–997
lifetime achievement and retirement, 1012–1014
family planning, 995
user-centered, problems with being, 992–994
word user as part of the problem, 992–993
Deaf and hard of hearing users, computing technologies for, 885
constructions, 1073–1076
display technologies, 887–889
multimedia interfaces, 888–889
signals, audible, 887–888
hearing and language, 886–887
hearing loss, 886
input technologies, 889–890
sign interfaces, 890
speech interfaces, 889–890
technology and communication, 890–892
Decision-support systems, human-centered design of, 570
case studies, 583–596
autonomous mobile robotic systems, human interactions with, 589–593
interactive critiquing as a form of decision support, 593–596
national air space system, distributed work in the, 584–589
decision-making tasks, human performance on, 577–582
cognitive engineering considerations, 582–583
errors and cognitive biases, 578–580
expertise, human, 580–582
design, approaches to, 575–577
task analyses and walkthroughs, cognitive, 575–577
work-domain analyses, 577
Design, 949, 1042
contextual, 949
contextual design overview, 950–961
contextual techniques into standard processes, integrating, 961–963
methodologies, general corporate, 962
programming, agile or extreme, 962–963
project scope, settling, 951–952
orientation to people's needs, 1043–1044
ambiguous and dynamic situations, management of, 1043
potential pitfalls, 1045
popularity, reasons for, 1042–1043
scenario-based 1041
user-centered, 993
Design, ethnographic approach to, 965
business strategies and models, 976–977
representations and models, 977
ethnographic methods, 969–970
access, gaining, 970
observation, 970
research planning, 969
ethnography, 982–985
experience models, 978
experiences, 976
field observations, structuring, 971
guiding principles and reference points, 978
ideas, evaluating and prioritizing, 977–978
inflection, postmodern, 968
innovation, supporting, 977
interactions, globally distributed, 975
interview as a communicative event, 972
interviewing, rules of thumb in, 972–973
mock-ups and prototypes, 980–982
observation and interviews, connections between, 973
self-reporting techniques, 973

organizational processes, 976
personas, 979–980
process models, 979
products, 976
observation, reason for, 970
relation to other qualitative approaches and perspectives, 982
relevance to design, 966–967
roots of ethnography, 967
principles, 967
representations and models, 978
types, 978
samples, 978
representations and models, value and function of, 977
researcher's observational role, 971
scenarios, 980
services, 976
social network analysis, 972
videotaping, 971–972
visual stories, 973
Weblogs, 973–975
artifact analysis, 974
data, qualitative and quantitative, 975
ethnography in a global context, 975
recordkeeping, 974
remote "virtual" observation, 974
working models of developers, enhancing, 977
Development process, sources of complexity in the, 499–502
maintenance, 502
problem finding, 499
problem formulation and requirements, 499–501
testing, 501–502
Digital divide, 764
roots, 765–772
anxiety and performance, 768–770
attribution, gender differences in, 770–771
low expectation, hazards of, 769
threat, stereotype, 770
Displays for selected applications, 222–225
mobile phones and handheld devices, 224–225
reality, 222–223
augmented, 222–223
visual, 222
Domains, importance of persuasion and motivation in, 135–136
persuasive technology, domains for, 135

E

Emotion, 78–79
definition, 78
emotions, basic, 79
Emotions and related constructs, distinguishing from, 79–80
mood, 79–80
sentiment, 80
Emotions, causes of, 82–84
appraisal theories, 82
contagion, 82, 84
emotional state, previous, 83–84
moods and sentiments, 83
needs and goals, 82
Entertainment interfaces and interactive products, emotion from, 694–695
affect's influence on product use, 695
game affect for entertainment access, designing, 695–697
new directions and open issues, 697–698
Evaluation, model-based, 1191
approaches, current, 1194–1195
architecture models, cognitive, 1194
effectiveness, reasons for, 1205–1206
GOMs models, 1194–1195, 1204–1206
task network models, 1194
model-based evaluation technique selection, theoretical basis for, 1195–1200
generativity, 1198–1199
modeling cognitive vs. perceptual-motor aspects of a design, 1197–1198
modern psychological theory, constraints in 1196–1197
psychological constraints, importance of, 1195–1196
role of detail, 1199–1200
model-based evolution technique application, practical issues in, 1200–1204
model validity, concerns over, 1203–1204
simulated device, creating, 1200–1202
task strategy, identifying, 1202–1203

F

Feedback, 167–168
correspondence, kinesthetic, 167–168
proprioceptive and kinesthetic, 167
snapping behaviors and active haptic feedback, 168
Fiero from hard fun gameplay, 686–687
player progress, enhancement, 687
players, mastery of challenge, 686–687
strategies and puzzles, creation of challenge with, 687
Financial methods, 1267–1268
dollars, raw, 1267–1268
internal rate of return, 1269
payback period, 1269
Finger/hand haptics, technologies in, 233–234
exoskeleton, 233
haptic interface, tool-handling type of, 233–234
Finger/hand output, technologies in, 234–235
Finger/hand output, technologies in, 234–235
FEELEX, 234–235
Full-body haptics, technologies in, 238–243
GaitMaster, 240–241
basic design, 240
control algorithm, 241
prototype, 241
locomotion interface, foot-pad based, 240–241
presentation of uneven surface, methods of, 240
robot-tile-based locomotion interface, 242–243
Circula Floor project, 242
infinite surface, method of creating, 242–243
treadmill-based locomotion interface, 238–240
basic design, 238
control algorithm, 239–240
mechanism and performance, 238–239
Torus treadmill, 238–240

G

Gameplay, amusement from, 692–694
creation of more emotion when playing face to face, 693–694

Gameplay, amusement from (*continued*)
emotion enhancement during group play, 694
social interaction, encouragement of, 692–693
Gameplay, easy fun, 687–690
curiosity's retention of attention, 687–689
game design, 689–690
Gameplay, relaxation from, 690–692
enjoyment, role in in work accomplishment, 691–692
relaxation, and player behavior, 690–691
Games, 1072–1073
planet saber, 363–364
Globalization, 357–363
advantages and disadvantages, 359
aesthetics, 363
color, 363
icons, symbols, and graphics, 361–362
language and verbal style, 363
layout and orientation, 361
typography, 362–363
appearance, 361
interaction, 361
mental models, 361
metaphors, 360–361
navigation, 361
technology, 360
user demographics, 360
case study, 363–364
critical aspects for globalization, 360–361
development process, 359–360
globalization, internationalization, localization, definitions of, 357–359
guidelines, 361–363

H

Haptic interface, application area for, 243–244
application area for finger hand/haptics, 243
3D-shape modeling, 243
art, 243
Haptic User Interface, 243
medicine, 243
application areas for locomotion interfaces, 243–244
Haptic interface, volmetric object-oriented-type of, 237–238
virtual clay volflex, 238
Volflex, basic design of, 237
Haptics and methods for haptic feedback, 230–233
force display, 230–231
exoskeleton-type, 230
object-oriented type, 231
tool-handling type, 230–231
perception and full-body haptics, 231–233
foot pads, 232
pedaling device, 232–233
sliding device, 231–232
treadmill, 232
proprioception and force display, 230
skin sensation and tactile display, 233
sensation, somatic, 230
HCI and usability, 641–654
HCI studies of privacy, 642–646
anonymity, 643
personality, 643–645
privacy, 641–642
security, 646–651
HCI in security, 648–650
security, control over, 647–648
usability challenges, 650–651
utility of security software and mechanisms, 647
trust, 651–654
design factors, 651–652
trust, factors influencing, 652–653
trust and personalization, 653
trust and risk, 653
HCI for kids, 793
children and the design process, 797–801
designing and testing with kids, methods for, 797–801
video, use of, 797
children design, difference of, 794–797
background knowledge, 796
dexterity, 795
interaction styles, 796–797
reading,796
speech, 796
designing and children, 794
technology for kids, genres of, 801–806
child safety online, 806
education, 802–803
entertainment, 801–802
educational technology, genres of, 803–806
HCI research, technological advancements in, 859–866
interfaces, adaptive, 863–866
interfaces, perceptual, 859–863
HCI, past systems in, 96–102
Cognitive Complexity Theory, 98–99
Collaborative Activation-based Production System, 99–100
Model Human Processor (MHP) and GOMS, 97–98
Soar, 100–102
HCI, sources of difficulty for, 498–499
computation, improvement of, 498
syntax and semantics, understanding, 498
Human computer interaction (HCI), future of cognitive architectures in, 110–111
architectures' focus on cognition and performance, 110
engineering demands reduction, 110
Human computer interaction in healthcare, 661
computerized healthcare applications, evaluating, 669–674
future opportunities and challenges, 674–676
adaptive HCIs, designing and utilizing, 675
e-Health, 675–676
new information, developing and utilizing, 675
patient-provider interaction, supporting/enhancing, 674–675
healthcare industry, 662–665
healthcare regulatory environment, 664–665
healthcare informatics, 665–669
consumer health information, 665
consumer/patient Web-based applications and e-Health, 666
decision-support systems, 668–669
health communication, interactive, 665–666
provider healthcare informatics, 666–667
Human effort identification, validation of, 152–154
error modes, 153
Human error, 148
classification, 148–150
error types, 150
errors, attribution to individuals, 148
Freudan slips, 148
predicting, 150–152

Human information processing approach, 44
  human performance as a function of several processing stages, 44
  provision of a direct link between basic and applied fields, 44
Human values, ethics, and design, 1239
  human values and ethics design, approaches to, 1245–1249
    computer ethics, 1245–1246
    computer-supported cooperative work, 1247–1248
    design, 1248
      participatory, 1248
      value sensitive, 1248–1249
    social informatics, 1246–1247
  human values with ethical import, 1249–1250
    accountability, 1253–1254
    autonomy, 1252–1253
    calmness, 1254–1255
    human welfare, 1250
    identity, 1254
    consent, informed, 1253
    ownership and property, 1250
    privacy, 1250–1251
    trust, 1252
    sustainability, environmental, 1255
    usability, universal, 1251–1252
  moral, philosophical, and psychological foundations, 1243–1245
    diversity, designing for, 1245
    epistemology, moral, 1244
    moral theory, 1243–1244
    variability and universality, moral, 1244–1245
  professional ethics, 1255–1257
  values, implication in technological design, 1240
    position, embodied, 1240–1241
    position, exogenous, 1241
    position, exogenous, 1241–1242
Human-computer interaction in aerospace, 721
  automation to the flight deck, introducing, 726–730
  autopilots, 726–727
    air-traffic control, early, 730
    role allocation, 727–730
  challenges, future, 735–737
    3× air traffic, technologies needed for, 736–737
    aerial vehicles, uninhabited, 737
    displays, three-dimensional 735
Hybridity and the third space, 1064–1066
  hybridity and HCI, 1064–1065
  participatory design as the third space in HCI, 1065
  participatory design's own third space, 1065

I

Information processing, attention in, 54–56
  automaticity and practice, 56
  attention, models of, 54–56
Information processing, memory in, 51–54
  factors affecting retrieval of earlier events, 53–54
  memory, 51–53
    long-term, 52–53
    short-term (working), 51–52
Information technologies, acceptance by older adults, 785–786
Information technology for cognitive support, 811
  cognitive impairment, 812–813
  Intelligence Quotient, 813–814
  people with cognitive impairment, interface design's support of 814–821
    aphasia, help for, 819
    attention, effects of, 814–815
    communication, augmentative and alternative, 816
    dementia, support for, 820–821
    memory loss and dementia, 815
    nonliterate users, support for, 819–820
    prostheses, cognitive, 816–817
    software as a cognitive scaffolding and a prompt for communication, 817–819
    visuo-spatial, iconic and verbal abilities, 815–816
  research methodologies, 823–827
    disabling environment, the, 825–826
    dynamic diversity, design for, 826–82
    user sensitive inclusive design, 826
  systems to support people with dyslexia, 821–823
    dyslexics, experimental text reader for, 822–823
    problems of dyslexia, computer aid for, 822
Information-processing methods, 45–47
  chronometric, 45–46
  psychophysiological and neuroimaging, 46–47
  signal detection and theory, 45
  speed-accuracy, 46
    characteristic curve, 46
Information-processing models, 47–48
  sampling models, sequential, 48
  stage models, discrete and continuous, 47–48
Input device, 162
Input devices, evaluation and analysis of, 165–167
  Fitts' Law, 166
  Hick's Law, 166
  Keystroke Level Model and GOMS analysis, 166–167
  Steering Law, 166
  Minimum Jerk Law, 166
Input technologies, 162–165
  input device properties, 162
    device acquisition time, 162
    dimensions, number of, 162
    gain, 162
    indirect vs direct, 162
    metrics, 162
    property sensed, 162
  input device states, 163–165
    graphical user interfaces, states and transitions at the core of, 164
  pointing devices, 163
    joysticks, 163
    mice, 163
    tablets, indirect, 163
    touchpads, 163
    touchscreens and pen-operated devices, 163
    trackballs, 163
Input, trends for, 171–172
Inspection methods, 1182–1184
  DARe model and the development lifecycle, 1184
  DARe model and the scoping of UIMs, 1182–1184
  Discovery and Analysis Resources (DARe) model of UIM effectiveness, 1182
Inspection methods, improving, 1184–1187
  analysis, 1186–1187
  improvements, 1185
    method independent, 1185
    method specific, 1185–1186

Interaction, modalities of, 169
input, 169
bimanual, 169
pen and gesture, 169
speech and multimodal, 169
whole hand, 170
devices, a society of, 171
tables and screens, multitouch, 171
technologies, background sensing, 170–171
Interfaces and agents, adaptive, 433
concepts, 434–435
future of user-adaptive systems, 454–455
growing need for user-adaptivity, 454
successful adaptation, increasing feasibility of, 455
information about users, obtaining, 449–451
low-level indices of psychological states, 451
naturally occurring actions, 450
previously stored information, 450–451
self-reports and -assessments, explicit, 449–450
signals concerning the current surroundings, 451
information acquisition, supporting, 440–446
helping users find information, 440–442
information presentation, tailoring, 443–444
products, recommending, 442–443
provision of information, spontaneous, 442
supporting, 444–446
collaboration, 444–445
learning, 445–446
special considerations concerning empirical methods, 451–454
adaptive and nonadaptive systems, experimental comparisons of, 453–454
individual differences, taking into account, 454
usability under realistic conditions, checking, 454
usage scenarios and user requirements, earlier studies of, 452
use of data collected with a nonadaptive systems, 451–452
Wizard of Oz studies, 452–453
work of human designers, comparisons with, 453
system use, supporting, 435–436
adapting the interface, 436–437
dialogue, controlling a, 439–440
interaction with the real world, mediating, 438–439
system use, helping with, 437–438
usability challenges, 446–449
controllability, threats to, 447
experience, breadth of, 448
obtrusiveness, 448
predictability and comprehensibility, threats to, 447
privacy, threats to, 448
tradeoffs, dealing with, 448–449

K

Keyboards and text entry, 168–169
mobile text entry, 168
memory, procedural, 168

M

Manipulation, 300–309
keyboards, mobile, 303–304
speech interfaces, 304–309
navigator wearable computer, 304–307
speech translation, 307–309
VuMan 3 dial pointing, 300–303
MDE for plastic user interfaces, 1117–1123
functional decomposition, 1118–1119
UI plasticity, models for, 1119–1121
Mental model, 64–67
artifact, representational, 68
homomorphisms, 66
isomorphisms, 66
metal content about domains, refocusing of attention on, 64
problem space and encoding, 65
situation model versus a text base, 67
text and other artifacts, 67–69
users' knowledge about systems, labeling, 64
Yoked State Space hypothesis, 68
Mental models, instructions for, 69–72
instruction, multimedia, 69–70
Mental models, shared 72–74
absence of increase in convergence of mental models, 73
mental models, measurement of, 72–73
role differentiation, 74
teamwork and collaboration, 72
Mobile interaction design in the age of experience ecosystem, 459
interaction, spheres of, 461–468
communities, 465–468
content, 464–465
mental models (and their representation), 464
humans and devices, 461–464
mobile experience, 460
content, communication, control, 460
interaction to experience, 460–461
networked ecosystems, interacting with, 460
Model-driven engineering (MDE), 1114–1117
metamodel, 1116–1117
transformations and mappings, 1117
Mood, causes of, 84
color, 84
other effects, 84
Multimedia user interface design, 393
background, cognitive, 395–397
attention, selective, 395–396
emotion and arousal, 396
learning and memorization, 396–397
perception and comprehension, 395
definitions and terminology, 394–395
design process, 397–410
attention, design for, 407–410
information architecture, 398–400
interaction and navigation, 405–407
media selection and combination, 400–405
users, requirements and domains, 398
Multimodal interface design, cognitive science underpinnings of, 421–425
multimodal interface, 425–428
design, future directions for, 427–428
difference from graphical user interfaces, 425
multimodal input, integration and synchronization characteristics of, 422–423
multimodal language, features of, 424–425
multimodal systems, architecture and techniques in designing, 425–427
multimodal interaction, differences in, 423–424
multimodal interaction of users, 421–422

Multimodal interface design, goals and advantages of, 417–420
terminology, 418
Multimodal systems, reasons for building, 414
multimodal interfaces, types of, 414–417

N

Networks as enablers, 267–270
applications, 269–270
Networks as mediators, 270–279
issues and problems, 270
media issues, 277–279
broadcast and prerecorded media, 278–279
compression, 278
interactive conversation and action, 277–278
jitter, 278
reliability, 278
sound and vision, 278
network properties, 270–274
bandwidth and compression, 270–271
encryption, authentication, and digital signatures, 273–274
jitter and buffering, 271–273, see also 278
latency and start-up, 271
quality of service, 273
reliability and loss, datagram, and connection-based services, 273
ownership, privacy and trust, 279
UI properties, 274–277
awareness, 276–277
coping strategies, 275
delay and time, 274–275
network transparency, 274
race conditions and inconsistent interface states, 276
timelines of feedback/feedthrough, pace, 275–276
Networks as platforms, 286–290
infrastructure, supporting, 289–290
awareness servers, 289
event/messaging systems, 289
notification servers, 289
resource discovery, 289
Web service and remote procedure call, 289–290
networked systems, architectures for, 289
shared objects, accessing, 286–289
editing, synchronous, 287–288
groupware undo, 288
locking, 287
solutions, real, 288–289
replication, 287
Networks as subjects, 279–286
network models, 280–283
addresses, 282–283
internetworking and tunneling, 281–282
layers, 280
peer-peer and ad hoc networks, 283
protocols, 281
routing, 282
network, 283–285
awareness, 284–285
confusion, 285
management, 283–284
networks, understanding and managing, 279–280
seamfulness and virtual locality, 285
Neuroscience, cognitive, 57
Neurotechnologies for assessing user state, 1222–1230
arousal meter gauge, 1226
body position/posture tracking, 1223–1226
Cognitive Monitor, 1230
EEG, 1229
EKG (ECG)-based arousal and cognitive workload gauges, 1223
electrodermal activity (EDA)/GSR-based arousal and cognitive workload gauge, 1223
Engagement Index gauge, 1227
Event Related Optical Signal, 1230
eXecutive Load Index gauge, 1226–1227
LM-ATL's EEG-based technology, 1229–1230
LM-ATL's/NIR approach, 1228
New Workload Assessment Monitor, 1227–1228
P300 novelty detector gauge, 1227
pupilometry, 1228–1229
stress gauge, 1226
Neurotechnologies, application of, 1230–1235
future directions, 1235–1237
controllers, robust, 1237
mitigation strategies, 1237
national and supra-national institutions, roles of, 1237
HCI system design and evaluation capabilities, improving, 1233
sytems, closed-loop, 1231–1233
training systems and operator selection, 1233–1235
Nonspeech sound and HCI, history, 248
sound, use in conveying information to computers, 248
visual and auditory feedback as tools for interaction, 248
Nonspeech sound in HCI, reason for use of, 250
annoyance, avoidance of, 250–251
sound, 248–249
advantages, 248–249
problems, 249
speech and nonspeech sounds for interface design, comparing, 249–250
Nonspeech sound presentation techniques, 255–259
auditory icons and earcons, comparing, 258–259
auditory icons, 255–256
design guidelines, 256
earcons, 256–258
design guidelines, 257–258
dynamics, 257
pitch, 257
register, 257
rhythm, 257
timbre, 257
Novel multimodal interface design, 420–421

O

Older adults and the acquisition of computer skills, 786
Older adults, use of computer technology by, 778–782
computer technology usage in everyday domains, 779–782
usage patterns and trends, 778–779
healthcare, 781–782
home environments, 780–781
work environments, 779–780
Older computer user, understanding the, 782–785
aging and abilities, 784
cognitive abilities, 785
motor skills, 784–785
sensory process, 784
elderly, today's, 782–784

Online communities, 603
  case studies, 612–617
    computer aided language learning communities, 612–613
    game communities and activity theoretical analysis, 613–6
  Computer Mediated Communication (CMC) and online communities, 604–608
  CMC and online communities, examples of, 605–608
    online communities, 604–605
  online communities frameworks and methodologies, analysis of, 608–612
    content and textual analysis, 610
    log analysis, 609–610
    query-based techniques and user profiles, 609
    social network analysis, 610–612
Organizational experience, 899–901
  children's art product manufacturer Website, 900

P

Perceptual-motor interaction in applied tasks, 37–38
  digital assistants, personal, 37–38
  eye-gaze vs. manual mouse interactions, 38
  remote operation and endoscopic surgery, 37
Perceptual-motor interaction, 28–30
  attention and performance, 30–37
    characteristics, 30
    shifts and coordinate systems, 30–31
    stimulus characteristics and shifts, 31–37
  human information processing and perceptual-motor behavior 28–29
  translation, coding and mapping, 29
Performance evaluation, 309–311
  perception, 310–311
  speech, dual purpose, 310
Performance, workload and stress, individual differences in, 127–129
  affective state differences, 128–129
    hedonomics and individual differences, 129
    narrowing and adaptive response, 129
  trait differences, 127–128
Personalization, portals and communities, 567–571
  communities, 570–571
  personalization, 568–570
    adaptation, methods and models of, 569
    authoring, 570
    systems, adaptive and adaptable, 568
    user modeling, 569–570
  portals, 570
Persuasion, 134–135
Persuasive technology, positive and negative applications of, 143–144
Persuasive technology, potential and responsibility, 144
Player experience framework, 685–686
  fun key models, 685–686
Problem solving and decision making, 56–57
  problem space, 56
    components, 56
Prototypes and the design process, 1020–1025
  alternatives, selecting, 1023–1024
  design space, 1021–1022
  design, participatory, 1021
  design, user-centered, 1020–1021
  ideas, generating, 1022–1023
  strategies, prototyping, 1024–1025
    prototypes, 1024–1025
      horizontal, 1024
      scenario-based, 1024–1025
      task-oriented, 1024
      vertical, 1024
Prototypes as design artifacts, 1018–1020
  evolution, 1020
  interactivity, 1019–1020
  representation, 1018–1019
Prototypes, design specification and implementation of, 235–237
  FEELEX 1, 235–236
  FEELEX 2, 236–237
  FEELEX, characteristics of the, 237
Prototypes, iterative and evolutionary, 1032–1036
  user interface, 1033–1036
    development environments, 1034–1036
    toolkits, 1033
Prototyping, rapid 1025–1031
  offline rapid prototyping techniques, 1025–1028
    mock ups, 1025–1026
    paper and pencil, 1025
    video, 1026–1028
    Wizard of Oz, 1026
  online rapid prototyping techniques, 1028–1031
    scripting languages, 1030–1031
    simulations, interactive, 1030
    simulations, noninteractive, 1029–1030

R

Reality and pervasive computing systems, prototyping, 1036–1037
Recognition-based systems, designing, 190–195
  context, importance of, 191–192
  costs, 190–191
  feedback, role of, 191
  generalization versus specialization, 193
  interaction, implicit and explicit, 191
  novel interfaces, evaluating, 194–195
  priori knowledge, 192–193
  traditional versus nontraditional interfaces, 193–194
Requirements analysis, 920–922, 918–919
  task analysis, contextual, 921–922
  user profile, 920–921
Requirements specification, motivating and justifying, 923–926
  task requirements, 924–926
  user requirements, 923
  work environment requirements, 923
Requirements specifications, introducing, 918–920
  design/testing development, 919–920

S

Safety and usability of interest, Measures of, 708
Searching and querying, 565–567
  query formulation, 565–566
  results list, 566–567
    document clustering, 567
    ranking algorithms, community-based, 566–567
    visual interfaces, 567
Sensemaking with visualization, 539–542
  crystallization, 539–540
  information visualization, levels for applying, 540–542
Sensors and sensing modes, 178–183
  brain interfaces, 183
  context, 182
  gaze and eyetracking, 181

gesture, 181–182
identity, 182
occupancy and motion, 179
position, 179–180
range sensing, 179
speech, 181
touch, 180–181
SHERPA, application to safety critical systems, 154–153
Signal processing, 183–187
classification and modeling, 185–187
Bayesian Network, 187
decision trees, 186
Hidden Markov Model, 187
independence, conditional, 186
Support Vector Machine, 186
feature selection, 184–185
preprocessing, 183–184
Sound, perception of, 251–252
attributes, 251
definition, 251
frequencies, humans' perception of, 251–252
loudness, 252
Spaces and places, 1066–1068
sitings, 1066–1067
workshops, 1067–1068
Speech technology, selecting, 383–386
Automatic Speech Recognition (ASR), 383–384
natural language processing (NLP), 385–386
natural language understanding (NLU), 385–386
NLU systems, capabilities and limitations of, 386
speech-synthesis software, current capabilities and limitations of, 385
Text-to-Speech (TTS), 384–385
Spoken interaction, crafting 386–389
challenges of speech, inherent, 387
correction strategies, 389
dialog styles, 387
errors, handling, 388–389
prompt design, 387–388
Spoken interface lifecycle, 383
Standards, rules and legal regulations, and health protection aspects, 222
Stress and cognitive workload, relationship between, 119
cognitive workload as a form of stress, 119
performance workload, 119
Stress research, traditional approaches to, 116
Stress, implications for researchers and practitioners, 129
Stress, mitigation of, 119–123
future research, problems for, 123–127
adaptation under stress model, 126–127
mental resources, 123–126
hedonomics, 122–123
motivation and skill development, 122–123
teletic work and intrinsic motivation, 122
teletic work and motivational affordances, 123
person, changing the, 120–121
personnel selection, 121
training/skill development, 120–121
task, changing the, 121–122
automation, adaptive, 121–122
display design, 121
Structures, narrative, 1068–1072
stories, 1068–1070
dramas and videos, 1071–1072
photographs, 1070
Survey design and implementation in HCI, 1151
survey design in HCI, 1156–1164
survey application domains, 1159
survey design methodology, 1159–1164
survey types, 1156–1159
survey implementation in HCI, 1164–1168
open vs. controlled implementation environments, 1164–1166
survey implementation, challenges in, 1166–1168
sample representatives, 1166

T

Tangible User Interfaces (TUI), 469
basic model, 472
from Graphical User Interface (GUI) to TUI, 470
GUI, 472
TUI, 472–485
URP, 470–472
Task analysis data, collecting, 934–938
field studies, 937
rapid task analysis, 936–937
remote task analysis, 937–938
site visits, conducting, 935–936
users and environments, selecting, 934–935
working with users, preparation for, 934
Task Analysis for Error Identification (TAFEI), application to interface design, 153–154
Task analysis studies, international, 938–940
data analysis and presentation, 940–946
data collection, 939–940
preparation, 938–939
Task analysis, defining, 928–930
principles, 929–930
Task analysis, planning for, 931–934
background research, 931–932
decision-making, 932–933
project plan, 932
reusable data, gathering, 933–934
sign-off and schedules, 932
useful and usable data, provision of, 932
Task analysis, selling, 930–931
resistance and dispelling, 931
task analysis, using opportunities to sell, 931
Tasks as stressors, 116
computer systems' imposition of stress on users, 116
computer systems' support people in stressful circumstances, 116
Tasks, users performance of, 702–708
cell phone problem, scope of, 704–706
crashes associated with telematics, 704
trips, 706–707
users profile, 707
vehicles, 707–708
Technology and sound production, 252–255
MIDI synthesis and sampling for auditory interface design, 254
sampling, 254
sample rate, 254
sample size, 254
sound synthesis and Musical Instrument Digital Interface (MIDI), 252–253, 253–254
sound, three dimensional, 254–255
Technology transfer, 1212
building trust, 1213–1214
confidentiality and non-disclosure agreements, 1216
deal, crafting a, 1216

Technology transfer (*continued*)
HCI tech transfer, application to, 1214–1215
internal and external technology transfer, 217
technology transfer as a social process, 1212–1213
Technology, persuasive, 137–140
computers as persuasive media, 138–139
cause and effect simulation, 138–139
environments simulation, 139
objects simulation, 139
computers as persuasive social actors, 139–140
attitudes and behaviors, modeling, 140
functional triad summary, 140
social rules and dynamics, leveraging, 140
social support, provision of, 139–140
computers as persuasive tools, 137–138
decision-making, triggering, 138
process, simplification, 138
self-efficacy, increasing, 137–138
tailored information, provision of, 138
Telematic design, 714–715
IVIS estimates, 714–715
IVIS yellow and red thresholds, 715
SAE J2365 calculations, 714
Telematics, design documents for, 710–714
alliance principles, 713–714
documents, types of, 710
ISO documents, 711–712
non-ISO documents, 710–711
major non-ISO telematics guidelines and recommended practices, 711
SAE J2364 (the 15–second rule), 712–713
Telematics, evaluation of, 708–710
evaluation contexts, 709
Testing and iterating, 389–390
computing, major waves in, 135
persuasive interactive technology, 135
Theoretical Frameworks, 116–120
adaptation under stress, 117–118
analysis, emergent unit of, 117
apparaisal theory, 116–117
cognitive-energetic framework, 118–119
Transfer functions, 167
devices, 167
absolute, 167
motion-sensing, 167
self-centering, 167

U

UIM samples, 1175–1177
evaluation, heuristic, 1175
walkthrough, 1175–1176
cognitive, 1175–1176
heuristic, 1176
Unified interface design, 1084
alternative styles, relationships among, 1087–1088
design problem, 1084–1085
method, 1086–1087
task hierarchies, polymorphic, 1085–1086
Unified interface engineering, 1088–1096
adaptability and adaptivity cycles, 1093
architecture, 1089–1093
contest parameter server, 1091
decision-making component, 1091–1092
dialogue patterns components, 1092–1093
user-information server, 1089–1091
containment hierarchies, polymorphic, 1095–1096
decision-making specification, 1096
dynamic user interface assembly, 1093–1095
Universal HCI accessibility and functionally illiterate populations, 871
Universal theory of persuasion, absence of, 137
Usability engineering, scenario-based, 1045–1054
activity design, 1048–1050
learning, case-based, 1056
problem analysis, 1046–1048
scenario-based design rationale as theory, 1056
scenario-based iterative design, 1054–1055
cases and object-oriented design, 1055
nonfunctional requirements, analyzing, 1055
personas and environment, 1054–1055
systems, evaluating, 1055
usability, evaluation of, 1052–1054
user-interaction design, 1050–1052
Usability inspection methods, (UIMs) 1172–1175
analyst preparation, 1173
candidate problem discovery, 1173–1174
candidate problems, confirmation and elimination, 1174–1175
definitions, 1172
need and rationale, 1172
usability inspections, logical structure of, 1172–1173
Usability testing, 1130–1136
basics, 1130–1131
data recording and analysis, 1133–1134
participants as end users, 1131
participants' performance of tasks, 1131–1133
usability, focus on, 1131
core issues, 1138–1145
ethics, 1144–1145
exploration, 1140–1141
testing with special products, 1143–1144
thinking aloud, 1139–1140
usability, measurement of, 1138–1139
usability testing, reliability of, 1141–1142
validity, 1142–1143
value, 1145
future directions, 1145–1146
objectives, 1145–1146
RITE model, 1146
testing, asynchronous, 1146
history, 1130
locations, 1136–1138
lab testing, 1136
onsite, 1137–1138
remote testing, 1136–1137
types, 1134–1136
informal vs. invalid, 1134–1135
prototypes in usability tests, 1135
Usable interfaces, visual design principles for, 329
charts, diagrams, graphics, and icons, 344–347
basic principles, 347
color, 347–348, 351–352
color, how to use, 347–348
guidelines, 346–347
icons and visual cues, 344–345
illustrations and photographs, 345–346
tables, charts, diagrams, 344
definition, 331

design process, the, 331–332
designer role, 332
experience, designing the, 350–351
medium use, 351
time, element of, 351
graphic standards, 348–350
information design process, 332–333
audit, 333
design development, 333
implementation and monitoring, 333–334
letterforms, variations in, 339
serifs, 339
stress, 339
thick and thin, 339
page design, 341–344
design of a page, building the, 341–343
page design techniques, 343–344
process of good design, the, 332
typeface size and selection, 337
families of type, 338
serif and sans serif, 337–338
typography, 335–337
typographic guidelines, 339–341
black on white vs. black and dark on light background vs. light on a dark background, 340
justified vs. ragged right (flushleft), 340
letter spacing and word spacing, 340
line length/column width, 340
line spacing/leading, 340
output device and viewing environment, 339–340
type, highlighting with, 340
typefaces, decorative, 340
typefaces, combining, 339
weight, contrast in, (boldness), 339
visual communication and organization, universal principles of, 334
visual design, 334–337
good design, criteria for, 335
principles, 334
tools and techniques, 334–337
visual language, 351–352
graphics/icons, 351
guidelines, 352
legibility, 352
metaphor, 351
navigational aids, 351
readability, 352

U

User experience (UX), boundaries of, 898–899
organizations involved in end-user relationships, 898
UX as context, 898
UX, elements of, 899
User interface plasticity, problem space of, 1108–1112
context and context of use, 1108–1109
ontological foundation, 1108–1109
system adaptation, 1110–1112
state recovery, 1112
UI distribution, 1112
UI migration, 1112
UI remoulding, 1111–1112
usability, 1109–1110
User tasks, composition of, 165
compound tasks and chunking, 165
elemental tasks, 165
User view, The, 901–905
UX of organizations, 902–905
brand, 902–903
relationships, 904–905
UX of products, 901–902
User-centered design in games, 741
game testing and evaluation, factors in, 748–751
challenge, 750
ease of use, 749–750
game designer intent, 748–749
pace, 750
games, defining, 742–743
game world, rules in the, 743
goals, difference of, 742–743
industry competition, intensity of, 743
need for games to challenge, 743
games, importance of, 741
players, diversity of, 741
game design, principles and challenges of, 746–748
challenges, identifying, 745
game design, role of technological innovation in, 747
multiplayer games, balancing, 747–748
perceptual-motor skill requirements, 747
players, rewarding, 746
skills levels, addressing, 745–746
social networking, enabling, 748
story, 746–747
game types, 744–745
platforms, 743
games, user research in, 751–757
approach, combined, 757
principles in practice, 751–754
survey techniques, 754–755
studies, formative, 755–757
User-centered design, empirical approach to, 1266
business strategies and models, 976–977
design focus, shifting, 975
diaries, 973
ethical issues, 968–969
ethnographic insights, application to design, 975–976
representations and models, caveat regarding, 982
UX and management, 911–914
development, agile, 912
UX, introduction into an existing process, 913–914
UX, examining, 905–911
command-referent pattern, 187–188
field observation, 907–909
focus groups, 909–911
organizational goals, identifying, 905–907
Xwand system, 187

V

Virtual environments, 621
application domains, 631–635
VE as a selection and training tool, 631–633
VE as a system acquisition tool, 634–635
VE as an entertainment tool, 633–634
design and implementation strategies, 626–629
cognitive aspects, 629–628
content development, 628

Virtual environments (*continued*)
products liability, 628
usage protocols, 628–629
health and safety issues, 629–630
cybersickness, adaptation, and aftereffects, 629–630
social impact, 630
system requirements, 622–626
hardware requirements, 623–625
software requirements, 625–626
virtual environment (VE) usability engineering, 630–631
presence, sense of, 630–631
usability techniques, 630
virtual environment ergonomics, 631
Visual displays, 202
basic principles and concepts, 202–209
display, 203
dimensionality
segmentation, 203
electronic information displays, 202
head-mounted displays, 203
image generation, 202
displays, 213–216
cathodoluminescent, 214–215
electroluminescent, 214
head-mounted, 204
human-related, 204–205
laser, 215–216
plasma, 213–214
electronic ink and electronic paper, 216–217
quality criteria, 205–210
brightness and glare, 206–208
contrast and gray scales, 208–209
display colors, 205–207
refresh rates, 209–210
resolution, 208
technologies, 210
cathode ray tube, 210–212
electronic information display technologies, classification of, 210–???
nonemitter/Liquid-Crystal Display, 212
Visual functions, 855–859
definitions, 855
epidemiology, 855–856
selected visual disorders, 856–857
visual functions and age, 858
visual functions, specific, 857–858
Visual performance, 217–222
visual performance, effects of specific display factors on, 219–221
visual evaluations, measures used for, 218–219
task demands, 218
Visual structures, composed, 531–534
double-axis composition, 532–533
mark and case composition, 533
recursive composition, 533–534
single-axis composition, 531–532
Visual structures, interactive, 534–536
attribute explorer, 536
dynamic queries, 534
extraction and comparison, 536
linking and brushing, 534–536
magic lens (movable filter), 534
overview+detail, 534
Visual structures, simple, 524–531
n-variables, 527
trees, 527–530
Visualization reference model, 519–524
data structures, 519
information visualizations, taxonomy of, 523–524
mapping data to visual form, 519
VUMAN 3 maintenance inspection wearable computer, 296–297

W

Wearability, design guides for, 297–299
attributes, 299
computer, use as body armor, 297
symbol, 297
Wearable computing CAMP, 297
factors consisting the CAMP framework, 297
Web, reasons for difficulty in using the, 560–563
browsing and linking, 560–561
user interface issue, 562–563Web search and query, 561

Ergonomics and Human Factors

The Human-Computer Interaction Handbook

"If you care about Interaction Design, you should own this book. Exhaustive coverage by world authorities. … one book that covers the gamut. Highly recommended, highly practical."
— Don Norman, Northwestern University and the Nielsen Norman group, *Author of Emotional Design and The Design of Everyday Things*

"Comprehensive and thorough coverage of all the important issues related to user interfaces and usability. A useful reference work for anybody in the field…"

— Jakob Nielsen, Author, *Designing Web Usability: The Practice of Simplicity and Prioritizing Web Usability*

Exploring the evolution in how people use and work with technology, the second edition of **The Human-Computer Interaction Handbook** captures the most important scientific and technical know-how in the field. It provides an updated, comprehensive overview of important research, including insights directly applicable throughout the process of developing effective interactive information technologies. It features cutting-edge advances to the scientific knowledge base and visionary perspectives and developments that will fundamentally transform the way in which researchers and practitioners view the discipline.

The second edition features contributions from a selection of eminent professionals in the field worldwide. Topics include sensor based interactions; tangible interfaces; augmented cognition; cognition under stress; ubiquitous and wearable computing; privacy and security. With contributions from over 130 researchers and professionals, over 5,500 references, 400 figures, and 100 tables, the book provides a wealth of data and a fresh perspective on the field. New topics and authors ensure the revision contains new information and insights, the latest in research and practice, while retaining its reputation for presenting authoritative information in an accessible manner.

CRC Press
Taylor & Francis Group
an informa business
www.taylorandfrancisgroup.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

ER9314
ISBN 0-8058-5870-9
90000
9 780805 858709
www.crcpress.com